AIRS -- AQS Modernization Proposal

System Management Plan for AQS

Prepared for

Information Management Group
Information Transfer and Program Integration Division
Office of Air Quality Planning and Standards
Office of Air and Radiation
United States Environmental Protection Agency

March 31, 1996

Table of Contents

1.0 Executive Summary

This report presents a Systems Management Plan for re-engineering a key subsystem of EPA's national air database, the Aerometric Information Retrieval System (AIRS).

The Systems Management Plan was developed to meet Agency Information Resources Management (IRM) requirements for documenting plans for new **Major Agency Systems** investments. The plan was undertaken in compliance with Agency System Development Life Cycle (SDLC) guidance. The proposed re-engineering of the Air Quality Subsystem (AQS) of AIRS qualifies as a **Major Agency System** because its scope of use is greater than the Office of Air and Radiation (OAR), including all 10 Regional Offices and the Office of Enforcement and Compliance Assurance (OECA).

This plan documents the following benefits from re-engineering AIRS-AQS:

Enhanced functionality and expanded database over the current AIRS-AQS. This will enable Office of Air Quality Planning & Standards (OAQPS) to achieve statutory and regulatory requirements that have not been addressed by the mainframe-based subsystem of AIRS.

Improved query, analysis, reporting, and access capabilities for EPA and external users of the subsystem.

System capability to support key indicators, one-stop reporting, and common sense initiatives that are advocated by the Agency Administrator. These benefits are attributable to the selection of the Agency's standard Relational Data Base Management System (RDBMS) for AIRS-AQS re-engineering.

Adoption of a standard, ETSD-supported information technology architecture using standard development tools, processing technology, and communications protocols.

Long-term cost reduction in comparison to continued operation of the mainframe subsystem and support of mainframe software maintenance costs.

Implementation of the client/server AIRS-AQS subsystem during FY 1998 (approximately 24 months from the completion of this plan).

Background

The OAQPS manages and supports AIRS on the National Computer Center's (NCC's) IBM mainframe using ADABAS data management software. Through its subsystems: the Air Quality

Subsystem, Air Facility Subsystem (AFS), AIRS Graphics Subsystem (AGS), and the Geo-Common Subsystem (GCS), AIRS provides a national repository for information about airborne pollution in the United States.

The goal of this re-engineering project is to implement a client/server architecture for AQS. OAQPS management selected AIRS-AQS to prototype AIRS client/server re-engineering following a thorough management assessment[1] led by the Information Management Group (IMG) of the Information Technology and Program Integration Division (ITPID). The assessment was initiated to determine the future direction for AIRS and to address requirements that have evolved since the implementation of the mainframe system in 1987. It included the collection and analysis of functional and technical requirements for the "future" AIRS and an analysis of alternatives to achieve the stated requirements.

The OAQPS consensus at the conclusion of the management assessment was to select AIRS-AQS for re-engineering. The benefits to OAQPS are reduced costs for operating and maintaining AIRS-AQS, the ability to expand the mainframe system to incorporate additional functional requirements and database expansion, and a flexible information systems architecture that is adaptable to changing mission and information requirements.

Characteristics of AQS

The mainframe AIRS-AQS contains ambient air pollution data collected by EPA and state and local air pollution control agencies from thousands of monitoring stations. AIRS-AQS also contains meteorological data, descriptive information about each monitoring station (including its geographic location and its operator), and data quality assurance/quality control information. OAQPS and other AIRS-AQS users rely upon the subsystem data to assess air quality, assist in Attainment/Non-Attainment designations, evaluate State Implementation Plans for Non-Attainment Areas, perform modeling for permit review analysis, and other air quality management functions. AIRS-AQS information is also used to prepare reports for Congress as mandated by the Clean Air Act.

Today, States and local agencies submit their data directly to AIRS-AQS, receive data validation reports, and correct errors online. Data retrieval is readily available to users through standard or ad-hoc reports. Data files can be extracted from the database for use with statistics and graphics applications to generate trend analysis and perform modeling routines.

Reason for Re-engineering

Several factors have contributed to OAQPS's need for a management assessment of AIRS. Developments in information technology (specifically, client/server architecture) have introduced

---

[1] The *Management Review of AIRS* included interviews and decision meetings involving the Immediate Office of the OAQPS Director, the Office of Enforcement and Compliance Assurance (OECA), and all OAQPS divisions: Emissions, Monitoring, and Analysis Division (EMAD), Emissions Standards Division (ESD), Air Quality Strategies & Standards Division (AQSSD), and ITPID.

and demonstrated the success of new system architectures by offering benefits not attainable through the IBM compatible mainframe architecture. For example, the client/server environment improves the effectiveness of users because it incorporates features that are commonplace to personal computer (PC) operations. The lack of a PC-centric interface has become a more prominent criticism of the current mainframe-based AIRS.

Agency funding and strategic IRM initiatives are factors that influence the cost and functional requirements of AIRS modernization. The advent of a Working Capital Fund (WCF) to finance mainframe system operations is likely to affect the cost burden of mainframe system owners and users. Such IRM initiatives as the Common Sense Initiative and One-Stop Reporting Initiative require the integration of Agency databases to support cross media analysis and to use key indicators that will be recognized by all national environmental databases. Also, outstanding requirements imposed by the Clean Air Act Amendments of 1990 for data collection and information management require expansion of AIRS databases.

In response to these demands, OAQPS recognized the need to identify and apply cost-effective and proven information technologies to develop a new generation of AIRS. Client/server and RDBMS technologies provide OAQPS with the opportunity to develop aerometric data management applications that have enhanced functionality, improved user interfaces, and more efficient applications development techniques than those traditionally employed in the mainframe environment.

The consensus decision in support of re-engineering was based upon requirements obtained during interviews for the *Management Review of AIRS.* These requirements were used to develop alternatives for the "future" AIRS, including two re-engineering options. ITPID/IMG presented the AIRS modernization requirements and alternatives to OAQPS management[2] along with estimates of costs to implement each of the alternatives. ITPID/IMG consolidated the comments of OAQPS managers and senior staff to achieve a consensus for AIRS modernization. The consensus was to re-engineer AIRS-AQS to a client/server architecture utilizing EPA's standard Relational Data Base Management System (RDBMS).[3] The consensus decision is documented in the *Mission Needs Analysis.*

AIRS-AQS Functional & Technology Requirements

The principal system requirements identified by the users, management, and support staff of AIRS are listed below.

They are summarized in ITAS 8-350, Deliverable 3-1, *AIRS Recommendations Briefing*, which was prepared to support the ITPID/IMG presentation.

The Enterprise Technology Services Division (ETSD) of OIRM has identified and published standard hardware and software products that it supports for Agencywide use. The ETSD-supported RDBMS is Oracle, as documented in ETSD publications: *Information Technology Architecture Overview* and *Information Technology Architecture Road Map.*

Functional Requirements

- Provide the functionality of the current AIRS-AQS
- Provide multiple levels and methods of data accessibility
- Provide a system that meets the needs of its users
- Store additional information required by regulation
- Provide data edit and validation checks
- Support changes in reporting requirements over time

Technical Requirements

- Provide a Graphical User Interface (GUI)
- Provide a technical environment that reduces the cost of using AQS
- Provide a system that is flexible and easily modifiable

The detailed requirements analysis stage will review each of these requirements to determine the feasibility for implementation with the re-engineered AIRS-AQS.

System Charter

The system charter is to design and develop a new Air Quality Subsystem that applies new technology to efficiently and effectively meet the needs of AIRS-AQS clients. The goal is to provide a system that addresses the following issues:

- Changing information technical requirements and cost
- Deficiencies in the current AIRS-AQS mainframe system
- Increasing mission requirements of AIRS-AQS clients, especially those within OAQPS

Projected Timeframe

The re-engineering effort for AIRS-AQS is planned for completion by April 1998. At that time OAQPS will have a fully functional and operational client/server-based RDBMS system, and the mainframe-based AIRS-AQS subsystem of AIRS will be retired.

The first phase of the System Development Life Cycle (SDLC) began in January 1996 with the establishment of an AIRS-AQS Re-engineering Team (ART)[4]. During this Initiation Phase, the team completed a high-level requirements analysis (ITAS 8-374 Deliverable 2-3, *Requirements Analysis for AIRS-AQS*) and an options analysis (ITAS 8-374 Deliverable 3-1, *AIRS Modernization Recommended Approach*) to identify for OAQPS management a recommended approach for AIRS-AQS modernization. The completion and approval of this System Management Plan represents the completion of the Initiation Phase.

As soon as practical, ITPID plans to begin Phase II, the Concept and Analysis Phase. It is projected to be a six month effort (see *Table 1. AIRS-AQS Re-Engineering Project Plan*) lasting until the end of FY 1996 (September 1996). The other phases of the SDLC timeframe are:

Definition and Design Phase - October 1996 to March 1997

Development, Conversion, and Implementation Phase - April 1997 to March 1998

Operations Phase - Beginning April 1998

System Category and System Scope

The AIRS-AQS Re-engineering Project qualifies as a **Major Agency System**, because of the scope of system use. AIRS-AQS is a mission critical system used by multiple Regions and OAR With respect to other criteria used to define system category, AIRS-AQS is well below the threshold established for a Major Agency System and does not have any impact on the Major Agency System classification.

As a **Major Agency System**, the System Management Plan for Re-engineering AIRS-AQS must be reviewed and approved by the Agency's IRM community.

The scope of the project is to re-engineer AIRS-AQS to operate within a client/server architecture. The system will be designed to operate independently of the other AIRS subsystems that will continue to run on the mainframe. Any components of the Geo-Common Subsystem (GCS) or the AIRS Graphics Subsystem (AGS) that are used by AIRS-AQS will also be re-engineered to a client/server architecture. The design of the system will encompass all components of the AIRS-AQS application, which includes the client (GUI) interface, the server functions, and communication processes. The functional and technical requirements that have been identified will be evaluated during the detailed requirements analysis stage for possible inclusion in the re-engineered system.

---

4 The AIRS-AQS Re-engineering Team will provide project management of the re-engineering effort. The ART is comprised of designated OAQPS staff representing IMG and the subsystem user community. The ART will seek technical support from other EPA organizations, such as ETSD, and will seek input from AIRS-AQS users during the detailed requirements definition and system design activities.

System Management Plan

The System Management Plan is designed to ensure the quality of the application systems that are developed and implemented by the EPA. This Systems Management Plan specifies the development life cycle methodology, information technology architecture, and data architecture that are proposed by OAQPS for re-engineering AIRS-AQS.

Life Cycle Methodology

The system life cycle methodology required for all EPA applications development identifies generic stages that typically occur in any system development project. EPA permits the use of any specific system life cycle methodology as long as the generic stages are included. The eight system life cycle stages recognized by EPA's IRM guidance are presented in this document under, *EPA Requirements for System Life Cycle Management.*

The technical approach recommended for developing the AIRS-AQS subsystem addresses the requirements analysis, design, and programming phases of EPA's SDLC. The recommended approach consists of two main elements: 1) a systems engineering approach for systems design and development, and 2) the data, process, technology, and network architectures that describe the key components of AIRS-AQS.

The systems engineering approach recommended for AIRS-AQS Re-engineering is a hybrid using components from other standard development approaches. The Zachman Framework (depicted in *Table 2*) will be used as the architectural checklist to ensure that all requirements are considered during the systems life cycle. Joint Application Design (JAD) and Rapid Application Development (RAD) methods will also be used as parts of the systems engineering approach.

Computer-Aided Software Engineering (CASE) tools are used within the software development life cycle with the goal of improving the efficiency of the development effort. Improvements are typically obtained from the use of CASE tools in the form of productivity gains, quality improvements, and increased maintainability. The proper use of CASE makes the system easier to maintain and enhances the system, as well as making it easier to integrate with other systems.

System Information Technology (IT) Architecture

A client/server architecture has been selected by OAQPS as the IT architecture for AIRS-AQS re-engineering. This IT architecture supports the key technical requirements identified during the *Management Review of AIRS.* It will facilitate the reduction in the costs of operating and maintaining AIRS-AQS, implement a graphical user interface, and establish a flexible systems environment that is adaptable for future growth and change. The detailed client/server model will be defined during the detailed design phase, based upon design criteria to be specified by the ART.

The proposed architecture supports EPA internal users (OAQPS, Headquarters, Regions) as well as external users such as States, local agencies, and the public. The AIRS-AQS re-engineering will use desktop PCs as the client workstation and Unix servers as the platform for the database server and SAS application server.

The conceptual system architecture capitalizes on the current Agency communications network used by OAQPS, Headquarters, and the Regions. It leverages the current explosion in Internet use as the preferred method for connectivity by external users. Internet will be used by the States and local agencies to access AIRS-AQS replacing dial-in and the limitations of a single point-of-presence on a closed SNA network. States and local agencies will use their own dedicated Internet connection or may subscribe to a dial-up Internet service provider[5]. The Internet provides AIRS-AQS with a common, open communications path for data collection and dissemination.

Software Development Environment

Oracle's RDBMS is an integral component of the AIRS-AQS software development environment. As a standard Agency product, supported by ETSD, Oracle provides a means to facilitate data sharing and integration with other Agency information systems.

Oracle products provide broad software development support and are available on Agencywide contracts. Oracle's RDBMS can be used as the repository for the AIRS-AQS data dictionary. Oracle's Designer/2000 is a CASE tool that supports EPA's full life cycle methodology. Developer/2000 will be used to generate forms and reports for the AIRS-AQS subsystem. Oracle's Discoverer/2000 is an easy-to-use query tool that allows users to develop their own ad hoc queries. The combination of the Oracle RDBMS, Designer/2000, and Developer/2000 provides a powerful, integrated, and effective software development environment.

Many other "query" applications are available to access the Oracle RDBMS. The RDBMS works effectively with many third-party software products that are in use within EPA, such as Lotus 1-2-3, SAS, WordPerfect, ARCview, and Microsoft Access.

Concurrent Users

An analysis of the current mainframe usage data indicates that there are 441 unique User ID's (UID's) that use accounts attributable to AIRS-AQS. The system configuration will initially provide concurrent software use licenses for 160 concurrent users. This capacity may be increased by upgrading the server software licenses at anytime during the systems life cycle.

Response Time

The system will be designed to provide a response time that is equivalent to or better than what is currently experienced with the mainframe version of AIRS-AQS. The most sensitive factor affecting response time in a client/server architecture is network loading. The application will be designed to minimize information traveling across the network by structuring server and client processing to maximize performance. Of the above factors involving response time, the load on the network is the item that is least controllable by the design of the system. The design will

---

[5] Most States have already established Internet connectivity and use such tools as World Wide Web servers to provide public information and access to state information resources.

limit the amount of data transferred between the client and the server in any transaction.

Security

ETSD Operational Directive Number 200.03, NCC Unix Security, defines a set of security standards for standalone or networked computer systems. This policy defines the responsibilities of all parties involved in providing a secure environment for the Unix platform.

The critical success factor for system security is controlling access to data. Confidential data is determined by statute and regulation, while sensitive data is determined by use. Data which is restricted by law or which is used for sensitive Agency functions (such as enforcement activities) must be protected against disclosure to unauthorized parties. The AIRS-AQS Subsystem will be designed to comply with Federal laws and guidelines affecting system security.

System security controls determine who may access the system, what resources they may use, and how they may use available resources. System security considerations include the following:

> **State sensitive data** will be protected from access by unauthorized individuals. The protection of state sensitive data has been accomplished in the mainframe AIRS-AQS by keeping it in separate files and allowing access only with the proper password.
>
> **Database security** is provided at the file and record level. Database security will be administered by a designated individual(s) responsible for the security of the system. The Oracle database security function is a sophisticated function meeting or exceeding the basic Federal and Agency requirements.

Operational Responsibility

The AIRS-AQS central subsystem server(s) will be housed at the NCC and operated by ETSD contractor support personnel. This arrangement is essentially the same as the operational responsibilities for the mainframe AIRS-AQS.

Server Hardware and System Software Requirements

A Digital Equipment Corporation Alpha model 8400 5/300 was selected on a preliminary basis as the database server system for the AIRS-AQS Re-engineering project. This hardware system satisfies high-level requirements developed to date and provides a basis for Cost/Benefit Analysis. The 8400 5/300 is a dual processor system that can provide the expected performance and meet the capacity requirements of AIRS-AQS.

Software requirements include the Digital Unix operating system (bundled with the Alpha server) and purchase of the Oracle RDBMS kernel software, parallel query option, 64 bit option, and SQL*Net TCP/IP. All of these products must be licensed for 160 concurrent users.

An equivalent server configuration will be required for the AIRS-AQS maintenance and support activities. The system does not have to provide the same performance and capacity as the production system but it should have identical system software configurations. Sofware licenses for 16 concurrent users should be sufficient for the system software.

Client Hardware and System Software

In order to fully utilize the potential of the new application, users will required PC clients with a minimum configuration of an Intel 486/DX33 processor, 8 MB of RAM, 150 MB of hard drive storage space, VGA video card, and TCP/IP network connectivity. In most cases equipment of this level is available to most of the users within EPA and external to the Agency. Generally, user expenses will be limited to AIRS-AQS client software and minor PC upgrades.

Network Capacity Requirements

Network requirements should remain nearly the same as they are today. There may be some increase in traffic due to new capabilities that will be added. The design of the system will focus on minimizing the impact on network traffic.

System Data Architecture

The database design and data flows will be refined during the detailed system design phase of the project. They are presented in *System Data Architecture,* in this document, to gain an understanding of the complexity of the legacy system and to provide a basis for estimating system development costs and hardware costs that are reasonably generated for estimated capacity and performance requirements.

Cost/Benefit Analysis

A Cost/Benefit Analysis was performed comparing the AIRS-AQS Re-engineering proposal with the existing mainframe AIRS-AQS. Costs for the re-engineering alternative were generated from hardware and software prices applicable to the Digital 8400 5/300 server system, Oracle RDBMS and development tools, and estimates of operating costs provided by ETSD. Mainframe costs were estimated and analyzed from the combination of mainframe CPU utilization data collected by the IBM MVS Integrated Control System (MICS) and Working Capital Fund cost data assembled by ETSD.

During the AIRS-AQS design and development phase (FY 1996-1998), re-engineering costs were in addition to those required to maintain AIRS-AQS mainframe operations. Through the projected implementation date of April 1998, these costs over 24 months (mid-FY96 to mid-FY98) were approximately $900,000 greater than the costs required to sustain mainframe operations.

However, as described in previous sections, the re-engineering alternative provides additional functionality and an expanded database compared to its mainframe predecessor. These benefits are realized upon implementation in April 1998. By FY 2001, the operating and maintenenace costs for the re-engineering alternative represent a savings against the costs for operating the mainframe AIRS-AQS. For FY 2001 the savings are $110,000 (approximately 5.5 percent less than mainframe susbsytem operations) and for FY 2002 they are $200,000 (approximately 10 percent less).

2.0 Background

This report documents the Systems Management Plan for re-engineering the Air Quality Subsystem (AIRS-AQS) of the Aerometric Information Retrieval System (AIRS) that is currently an IBM mainframe–based application using ADABAS data management software. The goal of this project is to implement a client/server architecture that reduces the costs of using AIRS-AQS and provides an environment that is flexible and adaptable to changing requirements. The re-engineering decision was reached by consensus with the Office of Air Quality Planning & Standards (OAQPS) Divisions participating in an AIRS Management Review Study and led by the Information Management Group (IMG) of the Information Technology and Program Integration Division (ITPID).

AIRS–AQS is used to assess the ambient air quality for the United States. It is an extensive repository of historical ambient air data that provides the EPA with a basis for trend analysis, assists users in air quality modeling efforts, and provides the necessary background for State Implementation Plan development for Non-Attainment Areas. AIRS-AQS accomplishes this level of functional support by utilizing an extensive database consisting of monitoring site description data, raw data for pollutant concentrations and meteorological conditions, summary data files for raw data and State/Local Air Monitoring System (SLAMS) data, and precision and accuracy data.

Factors Contributing to the Re-engineering Initiative

AIRS–AQS was first implemented in 1987 as a mainframe system in conjunction with the Geo–Common Subsystem (GCS). Several factors have contributed to the OAQPS need for a management assessment of AIRS. Developments in information technology (specifically, client/server architecture) have introduced and demonstrated the success of new system architectures by offering benefits not attainable through the IBM compatible mainframe architecture. Also, funding and strategic IRM initiatives of EPA present factors that will influence the cost and functional requirements of AIRS modernization.

Client Server Architecture

A client/server architecture incorporates graphical interfaces and allows easy definition of custom reports for systems users. These are some of the many features that improve customer satisfaction, system use, and data collection. Some AIRS customers (i.e., States and local air pollution control agencies) have installed Personal Computer (PC) and Local Area Network (LAN) based systems to manage aerometric data within their respective jurisdictions. Concerns expressed by other AIRS customers have convinced OAQPS that enhancements to, or modernization of, AIRS are desirable to maintain effective data collection and collaboration with external partners.

Funding Issues

EPA's Enterprise Technology Services Division (ETSD, formerly the National Data Processing Division) supports the central data processing and national data telecommunications of the Agency. ETSD is in the process of converting from an appropriations basis to a Working Capital Fund (WCF) basis for funding its operations. This change has the potential to increase costs to OAQPS and other EPA customers for mainframe systems support.[6]

OAQPS has directly funded development and maintenance costs of AIRS system components. These costs grew to more than $3 million per year[7] for FY92 and FY93 and are estimated at approximately $2.5 million for FY95.

In view of budget constraints prevalent in the Federal government, OAQPS is cautiously assessing its obligations and the alternatives available to manage and control costs, while maintaining mission support requirements.

Agency IRM and Strategic Management Initiatives

Agency IRM and Strategic Management Initiatives promote data integration, improved public access, Agency partnerships with environmental partners, and the effective use of information resource management (IRM) as a tool to achieve comprehensive environmental protection.[8] An example of Agency Strategic IRM Initiatives is the Key Identifiers Project begun in FY95. Representatives from EPA's Program Offices have been brought together in task forces to develop standards for identification and referencing of regulated and permitted facilities. These initiatives can best be realized by applying new information technologies, such as client/server, to information management requirements.

Description of Current System

AIRS is a computer-based repository for information about airborne pollutants in the United States and various World Health Organization (WHO) member countries. AIRS is installed on the IBM mainframe computer system located in the EPA's National Computer Center (NCC) in Research Triangle Park, North Carolina. AIRS consists of four subsystems: the Air Quality Subsystem, the AIRS Facility Subsystem (AFS), the Area/Mobile Source Subsystem (AMS) and the GeographicCommon , and Maintenance (Geo-Common) Subsystem (GCS). AIRS also

---

6 WCF charges for support services would be based upon mainframe CPU utilization and data storage requirements (DASD and tape volumes assigned to applications systems). ETSD's annual mainframe operations costs are relatively inelastic and there is concern that downsizing decisions by other mainframe customers would impact remaining customers resulting in higher rates for service.

Authorized FY93 contractor funding for AIRS development and maintenance support was $3,350,000. *Mission Support Review of AIRS*, Booz_Allen & Hamilton, Inc., July 6, 1992.

8 The source for Agency IRM strategic recommendations is *Interim Recommendations, Information Resources Management Planning Task Force*, Environmental Information and Assessment Committee, National Advisory Council for Environmental Policy and Technology, March 22, 1994.

includes the AIRS Graphics subsystem for creating maps and charts.

The Recommended AIRS Modernization Action Plan described in the *AIRS Recommended Action Report* advocates the modernization of a selected subsystem of AIRS. Because of the need for and importance of consensus to this significant OAQPS commitment, OAQPS managers decided that AIRS-AQS should be the target subsystem for modernization. This decision recognizes the consensus on AIRS-AQS functional requirements and desired enhancements, as well as unique problems with AFS data that require further discussion before a clear plan can be adopted for AFS modernization. Therefore, the focus of the remainder of this section will be on the current AIRS-AQS functionality and operating procedures.

Overview of AIRS-AQS

The AIRS-AQS consists of data derived from measurements of ambient pollutant concentrations in the air and meteorological conditions reported by thousands of monitoring stations operated by EPA and other national, state and local agencies. The AIRS-AQS also contains descriptive information about each monitoring station, including its geographic location and who operates it.

All AIRS/AQS data is stored in an ADABAS database management system on the IBM mainframe. The States submit their data directly to the AIRS-AQS, receive data validation reports, and correct errors online. Data retrieval is readily available to users through standard or ad-hoc reports. Data files can also be extracted from the database for use with statistics and graphics applications.

OAQPS uses this data to assess the overall status of the quality of the nation's air supply. The information is also used to prepare reports for Congress as mandated by the Clean Air Act. OAQPS also uses the data to identify localities where the air quality needs to be improved.

Types of AIRS-AQS Data

The AIRS-AQS database contains the following types of air quality data[9]:

- Monitoring Site Descriptions
- Raw Data
- Summary Data
- Precision and Accuracy Data

Monitoring site descriptions consist of information submitted by state and local agencies about the monitoring sites that provide data to AIRS. The information stored in the AIRS Site File includes site location (geographic coordinates, street address, city, county, state, AQCR, etc.), site operational dates, the organization responsible for monitor operation, and many other items.

Raw data are the individual values of pollutant concentrations or meteorological conditions measured at the monitoring sites and supplied to AIRS by the national, state, and local agencies that operate the monitors. AIRS contains three raw data files. The Hourly File contains data sampled at intervals of less than 24 hours (of which the 1-hour interval is most common). The Daily File contains data for sampling intervals of 24 hours or more (of which the 24-hour interval

[9] Information in this section was taken from the *AIRS User's Guide, Volume AA1,* (February, 1996).

is most common). The Composite File contains data from composite samples (multiple samples combined and analyzed as one). There are actually two sets of raw data files. One set holds public data, and the other holds private, or "secured", data that are available only to the organization supplying them to AIRS.

Summary data are derived from raw data. There are two summary files in AIRS: the Air Quality Summary File and the SLAMS Summary File. The Air Quality Summary File is derived entirely by AIRS software from information in the raw data files; there is no direct input to the file. Conversely, the SLAMS Summary File consists entirely of the annual summary data each SLAMS monitor is required to submit to the EPA in accordance with the Clean Air Act.

Precision and accuracy data provide information regarding the precision and accuracy of air quality monitors. The AIRS Precision-Accuracy Raw Data File contains values of pollutant concentrations representing a known concentration indicated by the monitor. The AIRS Precision-Accuracy Monitor Summary File contains quarterly summaries of the Precision-Accuracy Raw Data for the monitor. The AIRS Precision-Accuracy Reporting Organization File contains summaries of the precision and accuracy of groups of monitors (all those operated by a particular reporting organization).

The AIRS Site File, raw data files, and Air Quality Summary File contain all the data that resided in SAROAD, the predecessor of AIRS. Thus, the Site File includes information about discontinued monitoring sites as well as active sites. The AIRS-AQS raw data files and the Air Quality Summary File contain all values from SAROAD, some of which date back to 1957. The Air Quality Subsystem uses the Federal Information Processing Standards (FIPS) geographic codes. Therefore, site IDs were changed when the data were converted from SAROAD. The Precision-Accuracy Reporting Organization File contains data that resided in the Precision-Accuracy Reporting System (PARS), some of which date back to 1981.

Data Input and Update Procedures

Most of the air quality and precision-accuracy data in the AIRS-AQS come directly from the State and Local Air Monitoring Stations (SLAMS) and Natioinal Air Monitoring Stations (NAMS) air monitoring networks, which are operated by state and local pollution control agencies. A small amount of raw data comes from the EPA or private sources. In addition, AIRS-AQS includes air quality data submitted by various World Health Organization (WHO) member countries.

New air quality data are loaded into the AIRS-AQS and existing data are modified or deleted using 80-character transactions. There are transactions for site and monitor information, SLAMS summary data, and raw data. A local, state, or EPA organization submitting air quality data to the AIRS-AQS creates a transaction file on the IBM computer system at the EPA's National Computer Center. The organization submitting data uses AIRS-AQS software to load the transactions into a "screening file," check the validity of the transactions, and correct any errors found. A screening file is part of the AIRS-AQS database, and is used to hold AIRS-AQS transactions during validation. Each organization submitting data to the AIRS-AQS has one or more screening files for its exclusive use.

When the transactions in a screening file have passed validation checks, the organization submitting the data notifies the AIRS database administrator that the screening file is ready to be used for updating the AIRS database. The database administrator performs updates on a regular schedule (usually once per week) using all screening files that are ready at that time. The transactions used to update the database are automatically removed from the screening file by the update program; any transactions that had not passed validation checks or that had been excluded from update processing by the submitting organization remain in the screening file. To complete the update cycle, the database administrator notifies the submitting organization that the update has been done, and that the screening file is ready to accept a new set of transactions. (Between notification of the database administrator and completion of the update, a screening file is locked; nothing in it can be changed and nothing can be loaded into it.)

AIRS-AQS Data Retrieval

Most of the air quality data in AIRS are available to any person or organization with legitimate access to the EPA National Computer Center. (However, access to private, or "secured," air quality data is restricted by passwords so that only the organization submitting the data can retrieve them.) There are three ways to retrieve air quality data from the AIRS database: Standard Batch Retrieval, Online Browse, and ad hoc Batch Retrieval.

The AIRS-AQS Standard Batch Retrieval option generates reports that have been defined by NADB. These reports can be produced as printed reports and/or work files. The user specifies the criteria for data selection and sorting, and chooses options that affect the report format or content. The AIRS software accepts this information, automatically generates the required JCL, and submits a batch job to produce the report and/or work file requested.

The AIRS-AQS Online Browse option allow users to view requested data online. The user specifies the criteria for data selection and the AIRS software retrieves the data that meet the specified criteria and generates one or more NADB-defined screens of air quality information.

If the Standard Batch Retrieval and Online Browse facilities do not satisfy their requirements, users can define supplemental reports via the AIRS-AQS Ad Hoc Batch Retrieval option. To do this, however, users must have fairly good knowledge of the organization of the database, names of data fields, and so on.

Functional Requirements

Several functional requirements have been identified by the users, management, and support staff during the requirements gathering phase of this project. These requirements have been consolidated and are summarized on the following pages. The detailed requirements analysis stage of the project will review these requirements to determine their feasability for implementation with the re-engineered AIRS-AQS.

Provide the Functionality of the Current AIRS-AQS

The current functionality provided by AIRS-AQS must be replicated in the new system. The current functionality includes:

- Data editing
- Updating

- On-line browsing
- User interface
- Standard reports
- Ad hoc reports
- Graphics and mapping

AIRS-AQS is a successful sub-system that has served its users well in the past, has many users and has achieved a good reputation among its users. Functionality should not be reduced in an effort to reduce costs.

Provide Multiple Levels and Methods of Data Accessibility

The AIRS-AQS data repository must contain detailed information to support the modeling and analysis activities performed by some of its users. However, different levels of data can be made available to different types of users. For example, summary level information, which is what an estimated 90% of users want, can be made accessible to users via a World Wide Web Browser. Detailed data can be made available to a smaller group through the AIRS-AQS system interface or via a file transfer protocol (FTP).

Provide a System That Meets the Needs of its Users

The primary focus of AIRS-AQS should be to satisfy the needs of the EPA. However, data accessibility by secondary users such as the States, local governments, other government agencies, international organizations, and the general public need to be considered. It is envisioned that in the future fewer States and local governments will need to use the national system for reporting purposes due to implementation of their own systems. The system must accommodate infrequent users and organizations that need access to the AIRS-AQS national database.

Store Additional Information Required by the Regulations

AQS must have the ability to store and report information that is required by the regulations or needed to determine if the regulations are being followed. For example, the monitoring sites need to have a status that identifies designated co–located sites.

Additional status information also needs to be stored and available for on-line review. Examples of the additional information include:

- Annual system audit reviews of State QA programs
- Annual network review
- Participation in the National Performance Audit Program (NPAP) by Agency and/or site
- Identify co–located monitors and record the number of readings for precision checks

Time variable status information also needs to be stored in the database in a manner that associates it with a given time period. Examples include the time period that sites were designated co–located, when agencies implemented system audits, or when monitors were scheduled for NPAP audits and when the NPAP audits were actually performed.

Provide Data Edit and Validation Checks

Some of the data fields in the AIRS-AQS transactions need to be made mandatory so that the AIRS-AQS reports can meet the requirements of 40 CFR Part 58. Certain fields in the AIRS-AQS transactions have missing values that occur occasionally or continuously. For example, the number of readings taken field (PCIN) from the P&A Monitor Summary table are always blank for intermittent samplers used for PM10 and Pb. As a result, ad hoc reports that try to determine data completeness by monitor always show zero data completeness for PM10 and Pb. Another example is that the reporting organization (RONM) is occasionally missing from the P&A Summary table. When this occurs AIRS-AQS cannot properly summarize the data by reporting organization and may give misleading results for precision and accuracy by reporting organization. A systematic review of the AIRS-AQS database is needed to address this problem in a comprehensive manner.

Some additional examples of the types of data edits and validation checks that are needed include:

- The reporting organization (RO) should be mandatory for all data submittals to improve the accuracy of integrated data quality estimates. This will also improve the accuracy of roll-up reports to state, region, and national levels.
- Verify that the span concentration is within the specified range when data is submitted from automated analyzers for precision checks. This will help exclude invalid results that affect estimates.

Analyze Data Submittals for Completeness

AIRS-AQS must automatically perform a quality assurance review of the data submitted and produce a report that specifies what data may be missing, and to validate individual data elements such as allowed span and audit concentrations, missing limits or values, valid method codes, etc. This requirement is an extension of the previous requirement to provide data edit and validation checks.

Record Time-Related Information in the AIRS-AQS Database

The AIRS-AQS database needs to be expanded to include temporal information to document changes in monitoring networks, sites, and instruments. For example, the current AIRS-AQS system cannot report when data was entered or edited. This makes it difficult to check on the timeliness of data submittals by States and local agencies. The lack of temporal information also makes it difficult to tell how the status of a monitor or site may have changed over time. For example, if temporal information was recorded it would be possible to report changes in a site monitor status changing between NAMS and SLAMS.

Additional benefits of recording time-related information include the ability to:

- Identify sites and/or monitors that were or should have been operating during a given time period.
- Schedules can be developed and recorded in the database for network modification.

- Generate a data submittal timeliness and completeness report.

Reliably Handle Data From Ad Hoc Retrievals

Data that is retrieved via some types of ad hoc retrievals may provide inaccurate results, particularly when performing statistical analysis of the data. Methods are needed to automatically review ad hoc data and alert the user of any anomalies that may exist. For example, database entries with missing fields may make reports based upon the information incomplete or suspect. Manual methods are currently used to aggregate, classify, and review the data. However, these data review tasks can be tedious, time consuming, and are prone to errors. Any degree of automation that can be applied to these manual tasks will greatly enhance the value of the data and the effectiveness of the users.

Typically, the ad hoc reporting system either does not transfer records with missing or invalid data or it transfers the records without any warning to the user.

The following alternatives would improve the ad hoc reporting process:

- Notify the user with the number of records found, the number of records with invalid, blank, duplicate, or suspicious entries, and the number of records found but excluded from the report. Maintain the excluded records in a file that could be examined by the user.
- Modify the ad hoc reporting process to include all records, even those with invalid, blank, duplicate, or suspicious data.

Support Changes in Reporting Requirements Over Time

The system must support the existence of changes to the requirements in regards to what gets reported by the States. Due to changes in the requirements, previously submitted information may have missing items or may not conform to more recent validation checks (e.g. nonspecific method codes). AIRS-AQS must allow these changes in reporting requirements to be implemented with minimal impact on the integrity of the existing data.

Track Annual System and NPAP Audit Information

AIRS-AQS should allow information about system audits to be recorded in the system database. This information can be integrated with other QA related information such as when system audits were performed for the reporting organization, when NPAP audits were scheduled, and when actual NPAP audits occurred. Any of the information generated by the system audit could be recorded in the AIRS-AQS system database for online access and historical review of audits.

Enhance Existing Standard Reports and Provide Additional Reports

Enhancements to some of the existing standard reports can create improvements in the ability to assess the effectiveness of the ambient air monitoring programs. Examples of enhancements to the existing reports include:

- Generate summary reports at increased levels of resolution.
- Add more summary information to the reports.

- Allow reports to be generated for more flexible time periods.
- Add new data elements to the reports.

New standard reports that would be beneficial to the users include:

- A report that provides a summary of audit information for a site or monitor during a user–specified time period. Information included in the report would include summaries of NPAP audits scheduled, NPAP audits conducted, other on-site performance evaluations, and data from site reviews.
- A report that provides a summary of audits for a reporting organization, state, region, or the nation during a user–specified time frame. Information included in the report would include the number of NPAP audits scheduled, the number of NPAP audits conducted, and the number of system audits conducted by the EPA.

Integrate Ambient Air Data with Its Associated QA Information

The integration of air quality data with its associated QA information would provide real value to the AIRS-AQS data from a QA perspective. The data quality would be documented and accessible on–line by users when they are determining the suitability of data for particular purposes. This information would allow the user to determine if the data was collected and entered in accordance with EPA regulations and guidance.

Technical Requirements

Some technical requirements were identified by the users, management, and support staff during the requirements gathering phase of this project. These requirements have been consolidated and are summarized below. The detailed requirements analysis stage of the project will review these requirements to determine their feasability for implementation with the re-engineered AIRS-AQS.

Provide a Graphical User Interface

AIRS-AQS should include a graphical user interface that provides a screen display which makes it easier for users to run applications, specify information, and select options through the use of a mouse, icons, pull-down or pop-up menus, and other features. Graphical user interfaces provide standardized interface objects and interaction techniques that increase the useability of the application. Studies have shown that graphical user interfaces allow users to become familiar with applications in less time, reduce user frustration levels, and make users feel more comfortable with the system.

Provide a System That Reduces the Cost of Using AQS

The cost of using and maintaining AIRS-AQS on the mainframe is very high. An alternative technology, such as client/server, should be utilized to reduce the costs associated with AIRS-

AQS. A reduction of functionality within AIRS-AQS as a cost-saving option would be detrimental to the end users and reduce the usefulness of AIRS-AQS.

Provide a System that is Flexible and Easily Modifiable

AIRS-AQS should be built on an architecture that facilitates growth and change. In the past, OAQPS has struggled with proprietary systems that are difficult to integrate and update. With technology changing rapidly, it is undesirable to design and implement an inflexible system solution that will not allow change over time. An open systems environment that utilizes standards-compliant technology will be able to accommodate change without major conversion costs. In an open systems environment, hardware is not limited by vendor and the range of available software is substantial. In addition, standards-based communication products and protocols support interoperability.

One problem with the current AIRS-AQS system is that mainframe development efforts take too long and cost too much. There are currently over 1200 forms and programs in AIRS. Adding a data element to the database typically requires many of the forms and programs to be changed. A new technology, such as CASE tools, may provide a means of making modifications to the system in a cost-effective and timely manner.

Provide Access to AIRS-AQS Data via World Wide Web Servers

The use of the Internet and World Wide Web servers to provide access to information is growing at a rapid pace. AIRS-AQS should support the growth in this area by implementing the technology needed to distribute the AIRS-AQS data that is in the public domain from a Web Server.

Requirements for AIRS-AQS modernization are founded on the need to maintain, at a minimum, the current database and functionality while converting the application software to a client/server architecture. Additional requirements are identified to provide expanded functionality to AIRS-AQS through modernization efforts.

System Scope

The scope of the project is to re-engineer the AIRS-AQS subsystem to operate within a client/server architecture. The system will be designed to operate independently of the other AIRS subsystems that will continue to run on the mainframe. Any components of the Geo-Common Subsystem (GCS) or the AIRS Graphics Subsystem (AGS) that are used by AIRS-AQS will also be re-engineered to a client/server architecture. The design of the system will encompass all components of the AIRS-AQS application, which includes the client (GUI) interface, the server functions and communication processes. The functional and technical requirements that have been identified will be evaluated during the detailed requirements analysis stage for possible inclusion in the re-engineered system.

Assumptions

Several assumptions have been made during the development of this System Management Plan (SMP). These assumptions, which affect the system architecture, the system development life cycle methodology, and the users perception of the system include:

- New technologies are available that can be used to provide a more functional and intuitive user interface.
- The application of new technology to the AIRS-AQS system will improve the value of the services provided to those who use and rely upon the AIRS-AQS information.
- The operating costs for a client/server implementation of AIRS-AQS will be less than the cost of continued operation of the mainframe based AIRS-AQS.
- The AIRS-AQS subsystem and the data that it maintains is critical for monitoring and assessing the air quality of our Nation.
- The Oracle Relational Data Base Management System (RDBMS), which is the standard RDBMS for the EPA, will be used as the data repository.
- Oracle Designer/2000, which is a Computer Aided Software Engineering (CASE) tool, will be used to design the AIRS-AQS data model and functional process models.
- Oracle Developer/2000, which is an application development tool, will be used to develop the standard data entry forms and reports.
- Oracle Discoverer/2000, which is a suite of data query tools, will be used to provide some standard queries and provide users with the ability to create their own ad hoc queries.

3.0 System Charter

The system charter is to design and develop a new Air Quality Subsystem that applies new technology that efficiently and effectively meets the needs of the subsystem clients. The goal is to provide a system that addresses the following issues:

- Changing information technical requirements and costs
- Deficiencies in the current AIRS-AQS mainframe system
- Increasing mission requirements of AIRS-AQS clients, especially those within OAQPS

With the forthcoming of the WCF for funding mainframe system support, OAQPS recognizes the need to apply cost-effective and proven information technologies, that are supported by ETSD[10], to develop a new generation of AIRS. Client/server and RDBMS technologies provide OAQPS with the opportunity to develop aerometric data management applications that have enhanced functionality, improved user interfaces, and more efficient applications development techniques than those traditionally employed in the mainframe environment.

Mission Needs Analysis

In August of 1995, IMG initiated a project to identify user requirements and investigate how AIRS should be modernized to meet these new requirements. The first phase of the requirements analysis consisted of interviewing key personnel in OAQPS and the Office of Enforcement and Compliance (OECA). The interviews were structured to define the following :

- Scope and purpose of AIRS
- Clients and data
- Access requirements
- Functional requirement
- Performance requirements

The requirements gathered during the interviews were used to develop several alternatives for re-engineering AIRS. These alternatives were presented to OAQPS management (ITAS 8-350 Deliverable 3-1, *AIRS Recommendations Briefing*) along with estimates of costs to implement each of the alternatives. The alternatives were presented to OAQPS office and division directors by ITPID to solicit feedback and to initiate a consensus building process that would forge a recommended approach for AIRS modernization.

10 Modernization considerations assume the use of ETSD-supported technologies (e.g., Oracle RDBMS) that are recognized as "standards," as documented in ETSD publications, *Information Technology Architecture Overview* and *Information Technology Architecture Road Map.*

The results of the mission needs analysis, which was reached by consensus, identified that the AIRS-AQS subsystem should be the subsystem that is re-engineered to operate in a client/server architecture utilizing an RDBMS.

Sponsor Organization

OAQPS is the sponsoring organization for the AIRS-AQS re-engineering initiative. As the provider of a national air data system, OAQPS commissioned and directed a management analysis of AIRS to identify alternatives that would reduce the operating costs of AIRS and enhance the functionality to the AIRS user community through the use of new technology.

User Community

The user community serviced by the AIRS-AQS subsystem includes EPA clients, States, local air pollution control agencies, institutions, industry, and the public sector. This diverse set of users rely on AIRS-AQS to document ambient air conditions, to review and evaluate State Implementation Plans for criteria pollutant Non-Attainment Areas, and supporting air quality modeling activities.

Projected Time Frame

It is expected that AIRS-AQS re-engineering effort will provide a fully functional and operational system by April, 1998. At that time the existing AIRS-AQS mainframe-based subsystem will be retired and all users will perform their processing within the new client/server architecture.

The first phase of the system development life cycle has already started. The initiation stage began in January, 1996 with the establishment of the AIRS-AQS Re-engineering Team (ART) and the subsequent development of a project plan. A high-level requirements analysis was completed by the team (ITAS 8-374 Deliverable 2-3, *Requirements Analysis for AIRS-AQS*) and used to perform an options analysis and identify a recommended approach to management (ITAS 8-374 Deliverable 3-1, *AIRS Modernization Recommended Approach*). The concept and analysis stage begins immediateley after completion of this System Management Plan.

Table 1. AIRS-AQS Re-engineering Project Plan, identifies the major tasks associated with the project, the activities within each task and the estimated completion dates. A complete project plan is included in *Appendix A. Detailed Project Management Plan*, and provides much more detail than this summary table.

Table 1. AIRS-AQS Re-engineering Project Plan

Major Task	Activities	Estimated Completion
Initiation	Establish Core Team - IMG	1/18/96
	Prepare overall plan	1/31/96
	Recruit OAQPS team members	2/2/96
	Send memo to reporting organizations informing them of the project and plans	2/2/96
	Obtain Oracle training for IMG staff	2/12/96
	Issue WA for analysis, design, and development	3/16/96
	Perform the high-level requirements analysis	2/23/96
	Prepare options/recommended approach report	3/1/96
	Prepare System Management Plan	3/31/96
Concept and Analysis	Distribute end-user questionnaire via STAPPA/ALAPCO	3/31/96
	Conduct end-user requirements meeting	5/96
	Form an end-user advisory group	5/96
	Prepare detailed requirements analysis	8/96
	Operate information booth at STAAPA/ALAPCO meeting	8/96
	Design the prototype user interface	6/96
	Develop prototype	8/96
Definition and Design	Prepare data flow diagrams	10/96
	Design client/server model	11/96
	Design the information model	1/97
	Design the user interface	3/97

Table 1. AIRS-AQS Re-engineering Project Plan

Major Task	Activities	Estimated Completion
Development, Conversion, and Implementation	Build the database	4/97
	Develop the software	11/97
	Perform testing and acceptance	1/98
	Prepare user documentation	2/98
	Prepare operations/support documentation	2/98
	Develop system conversion plan	4/97
	Develop system conversion software and procedures	10/97
	Perform conversion testing and acceptance	1/98
	Establish production database	2/98
	Install production software	3/98
	Conduct training	3/98
	Convert data	3/98
	Establish user support	3/98
Operations	Begin operations	4/98
	Provide maintenance and support	ongoing
	Perform system evaluations	ongoing

A full description of these five major tasks is included later in this document in *Section 4.0 System Management Plan.*

System Category

The information system category is a classification assigned to a system based upon a combination of factors including the type of system, cost, and organizational scope in terms of use and funding. All systems developed within the EPA are classified as one of the following categories:

- Major Agency System;
- Major AAship/Regional System;
- Significant Program Office System; or
- Local Office/Individual Use System.

The system category applicable to the AIRS-AQS modernization project is **Major Agency**

System. The basis for this classification is primarily the scope of the system since it is a mission critical system used by multiple Regions and OAR. The estimated life cycle and annual costs for the system are well below the threshold criteria established for a Major Agency System and do not have any impact on the Major Agency System classification.

Based upon this system category classification, appropriate levels of management must review and approve or disapprove this System Management Plan (SMP). The levels of management that must review the AIRS-AQS Modernization SMP are:

EPA Executive Steering Committee for IRM

OAR Assistant Administrator, OAR Associate Administrators, Regional Administrators (all regions), Laboratory Directors, Headquarters Staff Directors, General Counsel, and the Inspector General

OAR Senior IRM Officials (SIRMOs)

Director of OIRM

- System Sponsor (OAQPS)

Review and approval of the SMP begins with the System Sponsors (OAQPS). After approval of the SMP by OAQPS it is routed to the Director of OIRM for review and approval. The OAQPS System Manager is responsible for obtaining the review and approval of the SMP at all levels of management. The SMP continues up the review/approval chain until it is approved by the Executive Steering Committee for IRM or is disapproved by any management review group. In the event that the SMP is disapproved, it may be modified and undergo another review/approval process starting at the System Sponsor management level.

Project Management

The AIRS-AQS Re-engineering Team (ART) will be responsible for the project management functions and providing guidance for all technical aspects of the project. The team is composed of selected OAQPS staff with a designated team leader. Additional EPA staff from other Offices and Divisions, such as the Enterprise Technology Services Division (ETSD), will provide information to the team from their specific areas of expertise. Some of the project management decisions that the ART will be responsible for include:

- evaluating the users requirements
- selecting the requirements to be implemented
- changing the system requirements
- overseeing the design and development of the system
- resolving unforeseen conflict in design
- specifying system objectives and policies

An advisory team of end-users will also be assembled to assist in detailed requirements definition and review the system design.

4.0 System Management Plan

The system management plan is designed to ensure the quality of the application systems that are developed and implemented by the EPA. This section of the document specifies the system development life cycle methodology, system information technology architecture, and the system data architecture.

Life Cycle Methodology

The development of successful information management systems has always presented a significant challenge for information systems staff and end-users. The current trend of migrating to a client/server architecture for systems can enhance the potential for attaining success if the software development approach exploits the flexibility inherent to client/server systems.

Client/server is a relatively new technology that is rapidly evolving to better meet the needs of its users and the IS staff that develop and support the systems. The move to client/server is typically motivated by the need for flexibility and scalability of information systems. Flexibility must be considered within the realm of both technological and business aspects. Technological flexibility includes the ability to easily change where data is stored and where processing occurs. Technological scalability allows changes in the size of the computing system in small, relatively inexpensive increments. Flexibility from a business perspective allows the system to be adapted for changing business rules and processes that intrinsically evolve over time. Scalability within a business framework supports additional users and increasing amounts of information over time.

Although information system developers must continually be concerned with changing technology, the real goal of developers is to contribute to the success of their users. A successful system implementation is based upon the accuracy and timeliness that the system meets the needs of the users. However, the users needs are not always easy to identify and they change as the business environment evolves.

A successful information system has the following recognizable characteristics:

Business driven - the users and management must direct the development of the system to ensure that it meets their particular requirements.

Rapidly developed - the system, or small components of the system, must be developed quickly and provided to users. This allows the users to provide feedback on whether the system is satisfying their requirements and allows the users to obtain benefits from use of the new system.

Flexible - information systems must continually change as user needs change and as users demand more from their systems.

Reliable - users expect quality and reliability from their systems. A successful system provides users a feeling of consistency and predictability throughout the application.

The goal of a system life cycle methodology is to provide a system that exhibits these characteristics. The methodology selected must use a systems re-engineering approach that will build upon the existing AIRS-AQS design and support the implementation of a new information technology architecture. Selection of the proper methodology and adherence to the methodology throughout the system life cycle is essential for culmination in a successful system.

EPA Requirements for System Life Cycle Management

The system life cycle management required for each system developed for the EPA is based upon a set of generic stages that typically occur in a system development project. Any system life cycle methodology may be used as long as the generic stages are included in the methodology selected. The eight stages of the system life cycle for EPA systems development are as follows:

- Initiation - a request for the development of a system to meet a need for information or to solve a problem for the individual making the request.
- Requirements analysis - determination of what is required to automate the function(s) identified by the organization.
- Design - the stage that specifies the automated and manual functions and procedures, the computer programs, and data storage techniques that meet the requirements identified and the security and control techniques that assure the integrity of the system.
- Programming - coding of the program modules that implement the design.
- Testing and quality assurance - ensuring that the system works as intended and that it meets applicable organization standards of performance, reliability, documentation, integrity, and security.
- Installation and operation - incorporation and continuing use of the new system by the organization.
- Maintenance and enhancement - resolving problems not detected during testing, improving the performance of the product and modifying the system to meet changing requirements. (Full scale enhancements require full life cycle analysis.)
- Retirement - the stage which ends use of the system.

Technical Approach

The technical approach for developing the system consists of two main elements:

- Systems engineering approach for systems design and development
- Data, process, technology, and network architectures that describe the key components of AIRS-AQS

The systems engineering approach is a hybrid evolutionary approach using components from other standard development approaches. The Zachman Framework, as shown in *Table 2, AIRS-AQS Architecture Framework,* will be used as the architectural checklist to ensure that all requirements are considered during the systems life cycle. Joint application design (JAD) and rapid application development (RAD) methods will also be used as part of the systems engineering approach.

Table 2. AIRS-AQS Architecture Framework

	Data	Function	Network	People	Time	Motivation
Scope	List of things important to AIRS-AQS	List of processes that AIRS-AQS performs	List of locations at which AIRS-AQS operates	List of organizations or agents important to AIRS-AQS	List of events significant to AIRS-AQS	List of goals and strategies for AIRS-AQS
Enterprise Model	Entity relationship diagram with AIRS-AQS entities and constraints	Process flow diagram with AIRS-AQS processes and resources	Logistics network	Organization chart showing organization units and work product	Master schedule of AIRS-AQS events and business cycles	Strategic plan with objectives and approaches
Information System Model	Data model with data entity and data relationships	Data flow diagram showing application functions and user views	Distributed system architecture	Presentation architecture with roles and deliverables	Processing structure	Knowledge architecture
Technology Model	Data design	Structure chart with computer function and screen formats	System architecture with hardware, software, and communi-cation line specifications	User interface architecture	Control structure	Knowledge design
Components	Data definitions and descriptions	Software programs	Network architecture showing node addresses and protocols	Security architecture	Timing definition	Knowledge definition
Functioning System	**Data**	**Function**	**Network**	**Organization**	**Schedule**	**Strategy**

Information systems development is an evolving process that requires specific considerations in order to be successful. A systems development life cycle methodology is a primary consideration. Examples of methodologies that have been used for systems development include the following:

- top-down approaches
- bottom-up approaches
- transaction-based approaches
- knowledge engineering approaches
- rapid prototyping approaches

- evolutionary development approaches

The use of each of these approaches has resulted in the implementation of successful systems. However, in an effort to use a "best of breed" methodology, components of several of these methodologies have been selected for use in the AIRS-AQS evolutionary approach.

The selected system development life cycle (SDLC), consists of five major phases. A brief description of each phase, and how it relates to the eight generic phases required for system life cycle management is as follows:

- **Initiation** - identifies the information management problem to be solved with a focus on the pertinent information, organizations experiencing the problem, time frame available for establishing the solution, and overall value of the solution. The System Management Plan is developed during this phase to provide guidance during the subsequent phases. Project organization and staffing to be used through the end of the system life cycle also begins during this phase and includes planning for staff training. This phase implements the *initiation* stage of system life cycle management as required by the EPA.

  Concept and Analysis - provides a high-level, comprehensive model of the solution to the problem that will guide the effort in subsequent phases. This phase defines high-level functional and data requirements, and evaluates alternative solutions to these requirements. The solutions address all aspects of the system, including the information to be processed, functional processing capabilities, hardware, software, and communications to be used. A demonstration application is developed during this phase to show existing and potential users an example of the new application. This phase implements the high-level *requirements analysis* stage of system life cycle management as required by the EPA.

  Definition and Design - provides a detailed description of the information and processing capabilities required of the system and a description of how the system will provide these capabilities. This phase defines the details of both manual and automated procedures required to provide a total system solution. This phase completes the *requirements analysis* stage and implements the *design* stage of system life cycle management as required by the EPA.

  Development, Conversion, and Implementation - builds the system according to the design and installs the system into a production environment where it is made available to users. Required data is loaded into the database, the existing database is converted, and training is provided for the users and system support staff during this stage. This phase implements the *programming, testing and quality assurance*, and the installation portion of the *installation and operation* stages of system life cycle management as required by the EPA.

- **Operations** - provides the full capabilities of the system to the end users and provides ongoing maintenance support. This phase includes system modifications, periodic formal evaluations of the system, and the ultimate termination and archiving of the system at the end of its useful life. This phase completes the operation portion of the *installation and operation* stage and implements the *maintenance/enhancement* and *retirement* stages of system life cycle management as required by the EPA.

The act of completing each phase of this SDLC process has not proven totally successful in ensuring that operational systems are fully responsive to user requirements. One of the primary causes for unsuccessful system implementations has been the lack of user involvement and feedback during the development and testing phases of the project. To overcome this situation, the AIRS-AQS approach utilizes the concept of rapid prototyping for building smaller, less complex versions of the system to demonstrate specific components of the system. The rapid prototyping could focus on the user interface, the integration of software applications into a comprehensive solution, or the feasibility of using a new technology to solve a particular problem. This approach allows the users to have beneficial use of individual components of the total system throughout the development process. For AIRS-AQS, the design process occurs in rapid series with an evolutionary development process that reuses previously developed system components. These components are expected to have real uses and stimulate user feedback through demonstration prototype versions. For example, a user interface is being developed early in the life cycle. AIRS-AQS users will be able to use the interface to perform limited real-world functions. Based upon user feedback, various iterations of the user interface will be produced with successively greater functionality.

Architecture Framework

Table 2 AIRS-AQS Architecture Framework, is a modified version of the Zachman Framework, which is an aid to the system development process. The following explanation of the Framework is taken from *Extending and Formalizing the Framework for Information Systems Architecture.*[11]

> [An Enterprise] contains entities, processes, locations, people, times and purposes. Computer systems are filled with bits, bytes, numbers, and the programs to manipulate them. If the computer is to do anything useful, the concrete things in the [Enterprise] must be related to the abstract bits in the computer. [A] framework for information systems architecture (ISA) makes that link. It provides a systematic taxonomy of concepts for relating [elements of] the Enterprise to the representations in the computer. It is not a replacement for other programming tools, techniques or methodologies. Instead, it provides a way of viewing a system from many different perspectives and showing how they are all related.
>
> The purpose of the [Architecture] Framework is to show how everything fits together. It is a taxonomy with 30 boxes or cells organized into six columns and five rows. (The sixth row is a result of the five above it.)...Flow charts, for example, may be suitable for describing the cell in the function column, components row in the Framework; and entity-relationship diagrams (ERD) may be acceptable for the data column, system model row. But the Framework shows how the cells in different columns and rows relate to one another.
>
> When applied to an information system, the word architecture is a metaphor that compares the construction of a computer system to the construction of a building. The Zachman Framework is an elaboration of the metaphor. It compares the perspectives in describing an information system to the perspectives produced by an architect in designing and constructing a building.

The Framework identifies that the information system development process requires several

---

11 Sowa and Zachman, *Extending and Formalizing the Framework for Information Systems Architecture,* IBM Systems Journal, Vol. 31, No. 3, 1992.

levels of architectures. As depicted in *Table 2*, the six Framework architectures are: *Data, Function, Network, People, Time,* and *Motivation*. The levels of the Architecture are *Scope, Enterprise Model, Information System Model, Technology Model,* and *Components*. Each Framework architecture (e.g. Data) develops as the process progresses along the Framework's life cycle from *Scope* through *Components*. Following is a description of each framework level, using a building construction project as an example. The first three levels provide a technology-independent model to accommodate emerging hardware and software development and software changes.

Scope. The first architectural sketch depicts in gross terms the size, shape, spatial relationships, and basic purpose of the final structure. In the Framework, it corresponds to an executive summary for the planners of AIRS-AQS and identifies what it will cost and how it will perform.

Enterprise Model. Next are the architect's drawings that depict the final building from the perspective of the owner, who will have to live with it in the daily business operations. They correspond to the Enterprise Model, which constitutes the design of AIRS-AQS by showing the interaction between the basic entities and processes.

System Model. The architect's plans are the translation of the drawings into detailed specifications from the designer's perspective. They correspond to the system model designed by systems analysts who must determine the data elements and functions that represent AIRS-AQS entities and processes.

Technology Model. The contractor must redraw the architect's plans to represent the builders perspective with consideration for the constraints of tools, technology and materials. The builder's plans correspond to the technology model, which must adapt the AIRS-AQS information system model to the details of the programming languages, databases, communications and other technology.

Components. Subcontractors work from shop plans that specify the details of parts or subsections. These correspond to the detailed specifications that are given to the information system developers who develop individual database systems and other components without being concerned about the overall context or structure of the system rows of the Framework.

Using the Framework checklist ensure that all aspects of the system are addressed from a total system approach and that all participants in the development and implementation of the system have a common reference point. This Framework has also been adopted for use by the EPA Office of Information Resources Management (OIRM) as the basis for an EPA-wide architectural approach. The Framework has also been successfully implemented for development of the Environmental Monitoring and Assessment Program (EMAP) Information Management System.

Design and Development Approach

Computer-Aided Software Engineering (CASE) tools are used within the software development life cycle with the goal of improving the efficiency of the development effort. Improvements are typically obtained in the form of:

- Productivity gains
- Quality improvements
- Increased maintainability

- Reusability benefits
- Overall consistency

Productivity gains are realized through reduced development labor which usually also reduces the elapsed time frame of the development project. The other improvements reduce the effort required to support the system throughout its useful life. The proper use of CASE makes the system easier to maintain and enhance, as well as making it easier to integrate with other systems. One of the major benefits of a CASE tool is its ability to perform an impact analysis prior to making any modifications.

Rapid Application Development (RAD) becomes a viable option within the life cycle methodology when CASE tools are used. A RAD methodology requires extensive user input throughout the development and generates iterative working versions of application components to provide a basis for end-user review and approval. This iterative development approach with end-user review allows design flaws to be identified earlier in the process, can be used to sell users and management on new concepts and results in reducing the costs and time for system development.

The CASE tool used must be repository-based and team-oriented. When the CASE tool is repository-based, the developers are able to reuse each other's work and share common definitions. This sharing and reuse supports rapid development and consistency within the system. Without a shared repository, effective reuse and sharing is impossible and consistency and quality checking is much more difficult and time-consuming. A team-oriented CASE tool allows sharing where it is useful, provides a secure environment for work-in-progress, supports repository browsing for component reuse, and maintains accurate documentation for the next developer who needs to use a component.

CASE tools typically support process modeling, systems modeling, systems design, and systems generation. Each modeling/design phase is used to create and refine information that describes the system and is stored in the repository. For example, organization units, processes, data flows, entities, relationships, business rules (constraints), and system development guidelines are all contained within the repository. Once in the repository, they can be used to create the database, forms, reports, and database trigger and stored procedure logic.

Process modeling is used to graphically model business processes that are performed to create or add some kind of value for the customers or end-users. Each business process has well-defined start and end points that are associated with an end-user. The models are usually developed in a hierarchical arrangement with major processes at the top and each having a number of sub-processes. The process models explain what occurs and ensure that all of the identified processes are implemented in the system. The models can also be used as a focus for people to agree on changing a process.

Systems modeling is used to create graphical function hierarchy diagrams, dataflow diagrams, and entity-relationship diagrams. The function hierarchy diagrams are used to decompose the business functions and show how the functions use data. The dataflow diagrams are used to show functions, datastores, data flows, and external datastores or functions. Dataflow diagrams can be used to identify data dependencies, system components, and the context of the project. The entity-relationship diagram identifies the information needs of the organization, the properties (attributes) of the information, and the relationships between the various types of information.

Systems design is used to create database definitions, business rules and procedures, screen prototypes, and workflow definitions. Logical and physical database schemas, GUI style preferences, application menu structure definitions, module data usage, screen prototypes, report prototypes, and stored procedures are typically defined during systems design. Data diagrams provide complete database designs. Module structure diagrams identify the functionality the system will present to the users and how the modules are arranged to support their tasks. Module data diagrams specify the internal structure of individual modules and are used to support rapid prototyping of screens and reports.

System generation is used to build databases and applications from information contained in the repository. Database objects, screens, and reports can all be generated from the information recorded by the other modeling/design tools. Preferences and templates can be specified prior to generation to allow a custom and consistent application to be developed.

The CASE modeling/design tools described above will be used at appropriate times during the various phases of the system development life cycle. The diagrams and reports that can be generated from the repository will be used as supporting information for many of the deliverables that are created during each phase.

Major Design Objectives

The design philosophy will be to provide the best service possible and will include :

Installation of a Graphical User Interface to enhance the user/machine interface.

Emphasis on system performance, accessibility, reliability, and flexibility.

Emphasis on the use of the capabilities of the servers including making full use of the server's ability to process information.

Emphasis on minimizing of the effect of the network (communications systems) on responsiveness of the system through efficient system design.

Provide a clear path for expansion of system capabilities and growth in data volume and number of users.

Minimize the effects of transition from the old system to the new platform.

Summary of Deliverables

Many deliverables must be prepared during the course of the system development life cycle. The purpose of the deliverables include describing requirements, documenting plans, creating user documentation, and development of the actual application. Individual deliverables are scheduled to be completed during various phases of the life cycle. Work on a deliverable may start in a prior phase but it is important to have the deliverable completed during the assigned phase. *Table 3. Deliverable Schedule*, identifies the deliverables that are required during each phase.

Table 3. Deliverable Schedule

Life Cycle Phase	**Deliverables**
Initiation	Needs Statement and Initiation Request Feasibility Study Cost/Benefits Analysis Risk Analysis Project Schedule
Concept and Analysis	Functional Requirements Analysis Functional Security and Internal Controls Requirements Analysis Data Requirements Analysis
Definition and Design	System, Program, and Database Specifications Data Management Plan Change Control Plan Data Administration Plan
Development, Conversion, and Implementation	Quality Assurance Plan Validation, Verification, and Testing Plan System Acceptance Plan User Manual Operations/Maintenance Manual Test Analysis and Security Evaluation Report Installation Conversion Plan Software Maintenance Plan Post Implementation Review Plan System Security Plan Disaster Recovery Plan
Operations	Evaluation and Assessment of Information System Obsolescence

System Information Technology Architecture

A client/server architecture will be implemented as part of the AIRS-AQS re-engineering. This information technology architecture supports many of the technical requirements of the AIRS-AQS re-engineering effort. For example, client/server will allow the costs of using AIRS-AQS to be reduced, will allow a graphical user interface to be implemented, and creates an environment that is flexible and adaptable for future growth and change.

Figure 1 shows the conceptual client/server architecture for the new AIRS-AQS operating environment. The architecture supports users that are internal to the EPA (OAQPS, Headquarters, Regions) as well as external users such as States, local agencies and the general public. Desktop personal computers have been selected as the client workstation for the AIRS-AQS application. Unix servers have been selected as the platform for the database server and SAS application server.

The conceptual system architecture capitalizes on the current Agency communications network that is used by the staff at OAQPS, Headquarters, and the Regions. It also leverages the current explosion in Internet use as the preferred method for connectivity by the States, local agencies, and the general public. TCP/IP has been selected as the communications protocol to be used over the EPA Network and from Internet-based users.

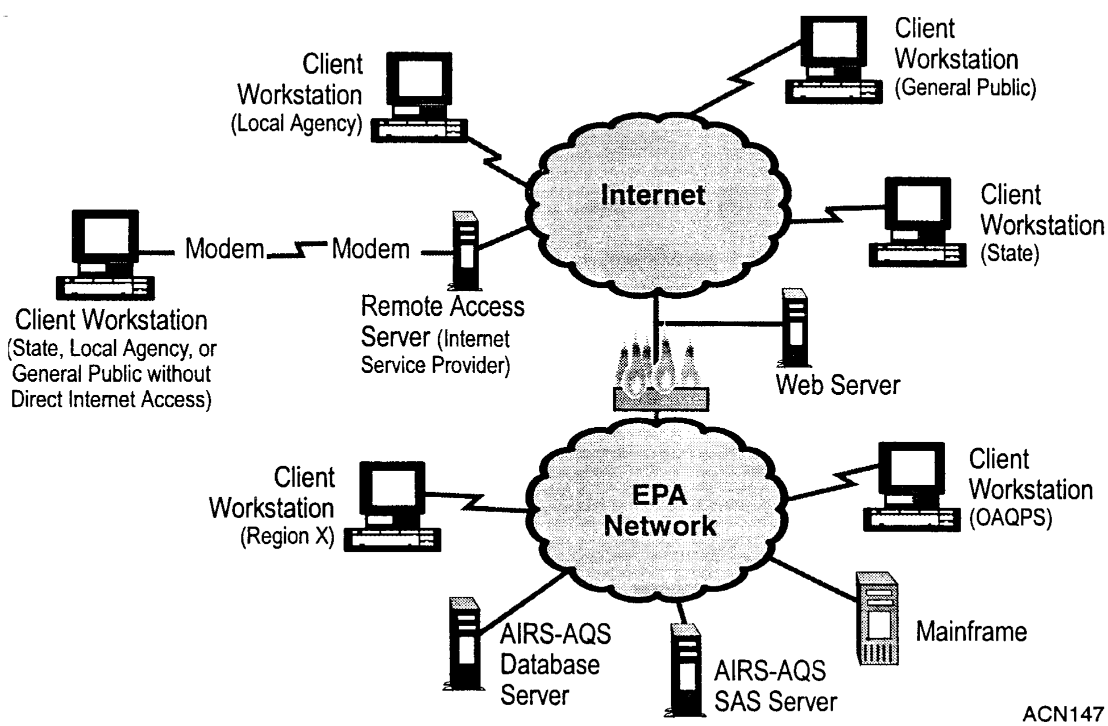


Figure 1. ***Conceptual Client/Server Architecture***

Client/server is the division of application processing (presentation services, application logic, and data management services) among front-end Client) workstations and back-end (server) processors. The clients always initiate a request while the servers wait passively for requests from clients. The conceptual architecture identifies the hardware components required to implement the client/server architecture. Although the hardware is important to the system design, the software architecture used for implementing the presentation services, application logic, and data management services is actually more critical for success within a client/server environment.

Presentation services provide the look and feel of the application to the users. The presentation services in a client/server application are usually a graphical user interface (GUI) that provides icons, check boxes, radio buttons, scroll bars, pop-up windows, etc. Presentation services are designed to make a system recognizable, familiar, and consistent across applications. They provide a graphical tool that enables information to be visually organized and provide an environment that allows total desktop integration of data among multiple applications.

Application logic is the way that the application processes data and responds to users online. The application logic can be subdivided into the categories of interactive application logic and data-intensive application logic. Interactive application logic, which provides data edits, selection lists, and integrity checks, can be performed by the client, by the server(s) or by both clients and servers. Data-intensive application logic, which provides services such as re-calculating summary totals, can also be performed by the client, by the server(s) or by both clients and servers. The partitioning of the application logic between the client and server(s) is a system design decision that affects the client workstation hardware configuration requirements and subsequent application performance. Communication with a server for every field validation adds an inherent network delay to the applications responsiveness while placing all of the data-intensive logic on the client may require client workstation configurations that are too expensive to be effectively utilized. A middle ground approach to partitioning the application logic is the best means of satisfying response time issues and hardware costs.

Data management services are typically provided by a database management system of choice. For AIRS-AQS, the Oracle Relational Data Base Management System (RDBMS) will be used to provide data integrity, minimize data redundancy, and provide distributed data access. The RDBMS also allows the data to be viewed globally by many applications rather than having the data be "owned" by a single application. This capability empowers users with the ability to expand the functionality of AIRS-AQS by using other applications that may be more appropriate for processing the data extracted from the AIRS-AQS database. Many commercial-off-the-shelf (COTS) applications, such as Access, Lotus, WordPerfect, ARCview, and SAS, can be used to connect to the database for user developed processes.

The types of client/server models, as shown in *Figure 2*, are directly related to the resultant partitioning of the presentation services, application logic, and data management services between the client and server. The distributed processing model places the application logic and data management on the server and essentially provides a graphical user interface to host-based processing that can be used by workstations with minimal resources. On the other end of the spectrum is the distributed database model that essentially runs the entire application on the users workstation and requires substantial processing power, memory, and disk storage. The optimal client/server approach exists between these two extremes. The application logic is implemented on both the client and server in a way that provides users with an interactive and easy to use application.

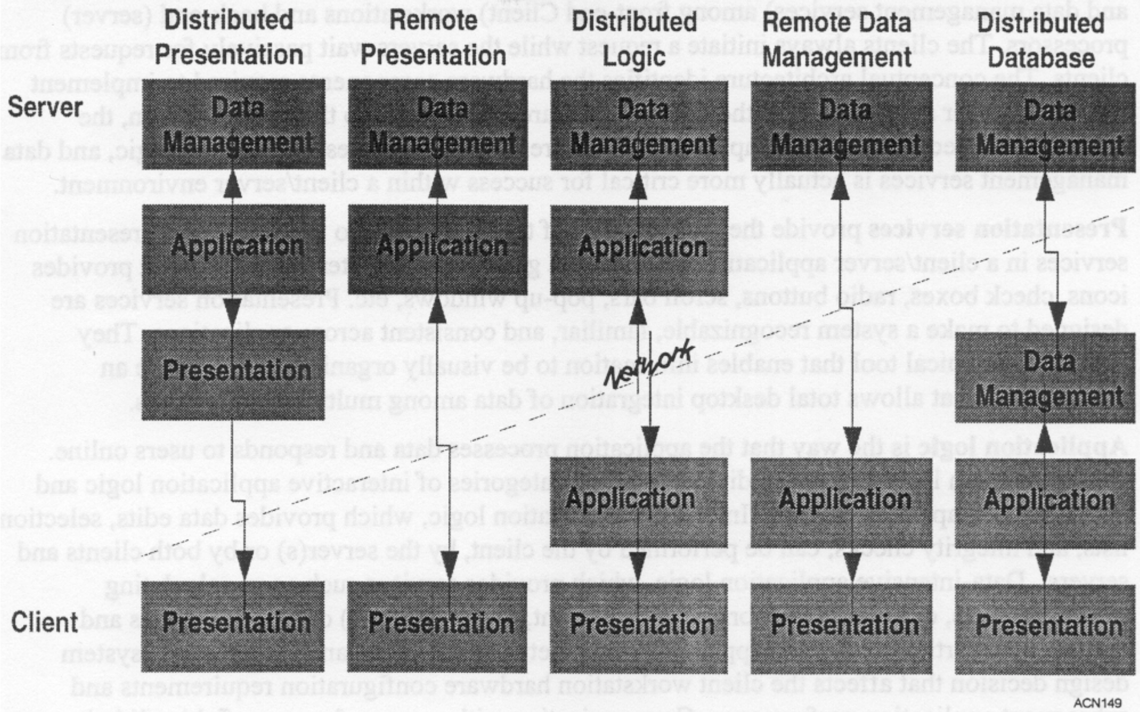


Figure 2. ***Types of Client/Server Models***

The client/server model will be defined during the detailed design phase and will based upon design criteria that has been specified by the project team.

Internet Access

Internet access to AIRS-AQS data and related information is one of the major requirements that will be addressd by the re-engineering initiative. The Internet provides a communication medium that supports access to the AIRS-AQS database by users that are not connected to the EPA network.

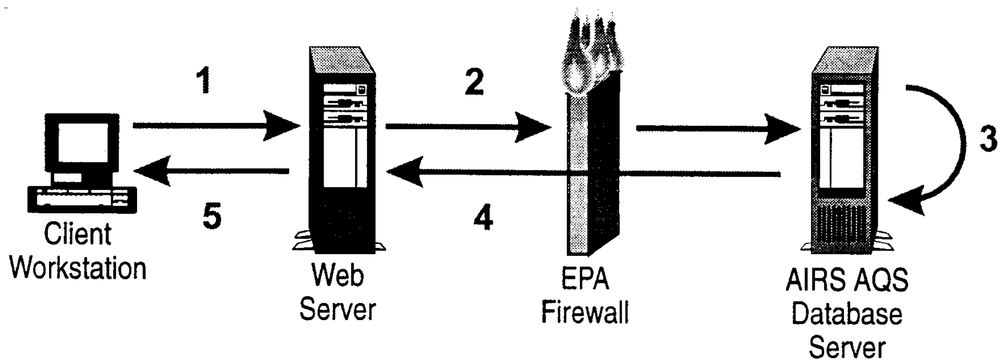
The Internet will be used by the States and local agencies to access the AIRS-AQS subsystem. The States and local agencies will use their own dedicated Internet connection or may subscribe to a dial-up Internet service provider. OAQPS will provide copies of the AIRS-AQS client application to all of the States and local agencies. This application will be designed to function over the Internet using the TCP/IP protocol. Other Commercial-Off-The-Shelf (COTS) applications, such as Lotus 1-2-3 and Microsoft Access, will also be able to access the AIRS-AQS database over the Internet.

The Internet may also be used to provide businesses, universities, and the general public with access to some selected AIRS-AQS information. An AIRS-AQS "home page" will be created on an EPA World Wide Web server that is outside of the EPA's firewall. The "home page" can be used to provide air quality datasets, reports, graphs, and any other pertinent static information to interested individuals. The Web server can also be used to process queries for specific information that are entered by the user. As shown in *Figure 3*, the query is sent to the database server and the response is routed back throgh the Web server for display to the user. This capability will allow users without access to the custom AIRS-AQS application to access the database. The types of static information and queries that will be available on the AIRS-AQS "home page" will be defined during the detailed design phase.

1 User enters a Query on a Web Page and sends it to the Web Server.

2 The Web Server parses the Query Message and sends a SQL transaction to the Database Server.

3 The Database Server processes the SQL transaction to get the query results.



5 The Web Server formats the query results into a Web page (HTML document) and sends it to the user.

4 The Database Server returns the query results to the Web Server.

ACN148

*Figure 3. **Query Flow on a Web Server***

Software Development Environment

Oracle's Relational Database Management System (RDBMS), EPA's stanadard relational database, is integral to the AIRS-AQS software development environment. The Oracle RDBMS has many features that support the mission of AIRS-AQS. These features include:

- Easy data access from GUI desktop applications;
- Data integrity can be built into the database with automatic enforcement;
- Designed for large volumes of data;
- Provides timely and continuous data access;
- Supports high transaction rates;
- Adaptable and scaleable to meet future needs;
- High reliability; and
- Minimizes network traffic.

The Oracle RDBMS can also be used as the repository for the data dictionary to be created during the system design phase. Oracle's Designer/2000 is a Computer Aided Software Engineering (CASE) tool that will be used for designing the system. Designer/2000 supports the full life cycle methodology described previously.

Oracle's Developer/2000 will be used to generate forms and reports for the AIRS-AQS subsystem. This tool suite is tightly integrated with the Oracle RDBMS and Designer/2000 to provide a very powerful, effective, and efficient software development environment. The custom AIRS-AQS application developed using these tools will be distributed to the Regions, States and local agencies.

Oracle's Discoverer/2000 is a set of tools that allow users to develep their own ad hoc queries. This tool suite is also tightly integrated with the Oracle RDBMS and provides users with an easy-to-use query tool. These tools may also be used to develop some "standard" queries that will be distributed as part of the AIRS-AQS application.

Many other "query" applications are available to access the Oracle RDBMS. Some examples of these third-party tools are:

- Lotus 1-2-3
- SAS
- WordPerfect
- ARCview
- Microsoft Access

These desktop tools, and others that individual users may prefer, can be used to query the database and return the results into the tool. For example, WordPerfect can be used to query some pollutant information from the database and include it into a report that is being written.

The Oracle software development environment also supports the use of 3GL programming languages such as C/C++, Fortran and Cobol. Oracle has developed extensions to these languages that supports connectivity to the Oracle RDBMS. This capability can be used to expand the functionality of the AIRS-AQS subsystem beyond the functions available within the Oracle development tools. For example, a module could be developed to read a data file submitted from a State, perform some validations on the data and then add the data to the Oracle AIRS-AQS database. The detailed design phase will identify any modules that need to be developed using a 3GL. It should be noted that Oracle's support for Cobol will allow existing Cobol procedures to be ported for use in the new AIRS-AQS subsystem. The ART will determine which 3GL to use based upon detailed requirements and the system design.

Ad Hoc Data Browsing Capabilities

Adhoc queries empower users with the ability to obtain information from the database without having to create a software program. The system architecture selected for AIRS-AQS exploits the use of ad hoc query tools to allow users to satisfy their own information needs. With the availability of ad hoc query tools, OAQPS can concentrate on maintaining the database and providing standard application functions. Individual users can develop their own special queries when they need the information. In typical legacy applications it usualy took several weeks or months for a special request report to be created for a user. This bottleneck in the software development group has been reduced by providing ad hoc query capabilities to the users.

Adhoc query tools are not a total panacea for satisfying users access to information. Standard reports are still typically needed to support the basic informational needs of the users. In any system there are some basic reports that are needed by a majority of the users. If these reports are not provided as part of the application system, the users will be forced to create their own using an ad hoc query tool. If there are 200 users there would be 180 different versions of the same basic report and 20 users wouldn't have time and/or knowledge to create the report.

SAS Analysis and Graphics Integration

There are several options available for integrating SAS analysis and graphics processing into the AIRS-AQS subsystem. The primary issues associated with the option(s) selected are cost and performance. The options available are:

- Users can run SAS on an application server and use an X-Window terminal emulation for presentation services.
- Users can run SAS on their client workstations and pull data to their workstation for analysis.
- Users can run SAS on their client workstations and submit remote procedures to a central SAS application server with the results returned to the client for presentation.
- Users can run SAS on the mainframe.

These options can all be used concurrently based upon individual user requirements and resources. This enables AIRS-AQS to provide a heterogeneous SAS computing environment.

Installing SAS on a central application server is very cost effective when compared to the cost of licensing SAS to every user that needs it. However, SAS running on an application server uses X-windows for presentation services. Therefore, the users must have an X-terminal or run X-terminal emulation software on their desktop workstations. X-windows, which provides a GUI, suffers from performance problems when used over a Wide Area Network (WAN). All mouse movements and screen updates must be transferred between the workstation/x-terminal and the server and the performance is directly affected by network communcation speed and activity.

Some users may prefer to license SAS to their workstation rather than use the central SAS server. The users can dynamically pull data from the AIRS-AQS database with SAS/Access and perform the analysis on their workstation. However, this technique can potentially move large amounts of data over the network and degrade the network performance.

Another option for users with SAS at their workstation is to submit a remote procedure to the central SAS server where the analysis is performed. The results are then returned to the users workstation for presentation. This option eliminates the transfer of large amounts of data over the network and allows the powerful SAS server to perform the analysis.

The last option is for users to run SAS on the mainframe as is currently done. This requires mainframe processing charges to be incurred but provides consistent performance and exploits the use of the mainframe SAS license. Users must have a 3270 terminal connection or use 3270 terminal emulation software.

It is envisioned that the AIRS-AQS data will periodically be exported from the Oracle database and loaded into SAS. The export file from the AIRS-AQS database can be used to update the mainframe SAS, central server SAS, or workstation versions of SAS. SAS users will have the option of using the most recent "snapshot" of the AIRS-AQS data that has been loaded into SAS or use SAS/Access to get the current data from the AIRS-AQS database.

Data Transmission from the States and Local Agencies

The States and local agencies collect the air quality data from monitoring sites and must periodically transfer the data to the AIRS-AQS subsystem. There are several options available for transmitting and loading the data within the new client/server architecture. These options include:

- Send data files over the Internet using FTP.
- Transfer data files via modem using a file transfer program such as CrossTalk or Kermit.
- Use Oracle's SQL*Loader utility to insert records directly into the AIRS-AQS "screening" database from the client workstation.
- Use an AIRS-AQS custom application to insert records directly into the AIRS-AQS "screening" database from the client workstation.
- Use SAS, Lotus or some other desktop tool to insert records directly into the AIRS-AQS "screening" database from the client workstation.

The method(s) to be used will be determined during the detailed requirements analysis and system design phases of the project. Each option has varying amounts of user interaction and data validation capabilities.

Options for Non-interactive AIRS-AQS Functions

Non-interactive AIRS-AQS functions, such as loading, browsing, editing, deleting, and data retrieval can be implemented using any of several different options. Each option has varying amounts of user interaction and client workstation processing requirements. The options available for non-interactive functions include:

- Use an AIRS-AQS custom application function to submit a batch job request to the database server. The batch job could be implemented as a stored procedure on the database server or could be a Unix script that is executed by the operating system.
- Use an AIRS-AQS application that runs in a secondary window on the client workstation. This will allow the user to validate and correct data files and view the results dynamically and/or after the completion of the processing. This method places the task of validating, correcting and loading the data on the user rather than the AIRS-AQS administrative staff.
- Use another application, such as SAS, to invoke a stored procedure on an application server.

The method(s) to be used will be determined during the detailed requirements analysis and system design phases of the project.

Options for Interactive AIRS-AQS Functions

Interactive AIRS-AQS functions such as online browsing, correcting, and on-line data retrieval can be implemented using several methods. These options exploit the openness of the system architecture and allow users to access the AIRS-AQS database in many different ways. Some of the options available include:

- Use custom AIRS-AQS functions that have been developed with Developer/2000.
- Use Oracle's Discoverer/2000 for ad hoc and pre-defined queries.
- Use SAS to create datasets for analysis from the AIRS-AQS`database.
- Use ARCview to access the AIRS-AQS database and create maps corresponding to the data extracted.
- Use other desktop applications, such as Microsoft Access, to gain interactive access to the AIRS-AQS database.
- Use Web Browsers, such as Netscape Navigator, to access the AIRS-AQS database through a Web server.

The method(s) to be used will be determined during the detailed requirements analysis and sy stem design phases of the project.

Concurrent Users

Analysis of the current mainframe usage data indicates that there are approximately 441 unique User ID's (UID's) that use computer accounts attributed to the AIRS-AQS subsystem. The system configuration will initially provide concurrent software use licenses for 160 concurrent users. This capacity may be increased by upgrading the server software licenses at anytime during the systems life cycle.

Response Time

The system will be designed to provide a response time that is equivalent to or better than what is currently being experienced with the system. Response time is based upon a number of factors:

- The amount of information traveling the network
- The load on the network
- The performance capabilities of the server
- The performance capabilities of the client workstation

Of the above factors involving response time, the load on the network is the item that is least controllable by the design of the system. The design will limit the amount of data transferred between the client and the server in any transaction.

Security

The plan for the application of security for any system begins with a management policy statement. It is from this statement that the level of security applied to a computer system is derived and implemented.

The Management Security Policy Statement

It is the policy that all data managed by the system be protected from unauthorized or unintentional modification. It is intended that most information be available to all who have need of this type of data.

There are some data in the system which are classified as state sensitive and are restricted to those who are responsible for the information. This data must be given special protection against unauthorized disclosure.

Operational Directives

The U.S. Environmental Protection Agency ETSD Operational Directives Manual contains a policy which address Unix based computing platforms. This is policy number 200.03 titled NCC UNIX SECURITY, and dated 12/2/93. It defines a set of security standards for standalone or networked computer systems. This policy defines the responsibilities of all parties involved in providing a secure environment for this platform. Section 7 of this directive provides additional references which should be reviewed. The following sources are to be used as references in the final design and implementation of the security module.

- OMB Circulars A-76, A-123, and A130. Which support setting the guidelines and procedures at the operational Levels.
- U.S. Environmental Protection Agency Information Security Manual
- The Computer Security Act of 1987 (public law)

Security Technology Overview

The essence of system security is controlling access to data. The controls determine who can access the system, what resources they may use, and how these resources may be used. To properly control access and effectively prevent compromise of the data, the system must perform the following functions:

- User identification
- User authentication
- Login control
- Password management
- Authorization
- Delegation
- Journaling

User identification names the subjects—interactive users, batch jobs, or application programs that can log onto the system to access application data, devices, or services. Users and programs log in through a user name or code that uniquely identifies each. If the user name is not listed in the login control files then access to the system is denied.

User authentication ensures that users are really who they claim to be by associating the user name with a password. Passwords provide a reliable method of identifying the user and ensuring access authority.

The login control function specifies the conditions users must meet before gaining access and grants access when those requirements are fully met. The security system can also restrict access based upon the type of terminal or time of day or day of week. In addition a system may send messages to the system operator and disable accounts when a break in is attempted.

Password management prevents individuals from guessing or obtaining passwords which do not belong to them. It should contain such functions as:

- Logging unsuccessful login attempts
- Requiring users to periodically change their password
- Specify a minimum length of a password
- Suppressing the view of the password to prevent it from being read by others
- One-way encryption of the password in the system's protected password file

Authorization is a function which identifies the system resources to be protected and describes who can use them. Protection is typically applied to application functions, files, tables, and utilities. Users can have any combination of read, write, execute, delete, alter, or control access depending upon their requirements.

The **delegation** function allows users to assign authorization to system resources to other users, such as managers, providing access to system resources to those individuals within their workgroup. A manager can only delegate access to resources that are authorized to the manager. The manager may also revoke such delegations.

Journaling is the function which records system activities in a log. This allow the security administrator to be able to:

- Identify access violations
- Determine security exposures
- Track the activities of selected users
- Adjust the access control measures to changing conditions

State Sensitive Data

State sensitive data has been identified and will be protected from access by unauthorized individuals. The protection of data classified as state sensitive has previously been accomplished by keeping it in separate files and allowing access to this data only with the proper password. This process will continue if the need continues to exist.

Data Base Security

In general, the database engine will provide file and record level security and will be administered by the individual(s) responsible for the security of the system. The Oracle database security function is a sophisticated function meeting or exceeding the basic requirements discussed earlier. The Oracle database contains a list of valid database users. To access a database, a user must run an application and connect to the database instance using a valid user name defined in the database.

Each user is authenticated by a password and access to the system is controlled through the use of user profiles, privileges, and "roles." A profile is a named set of resource limits. Oracle limits a user's use of database and instance resources to that described in the profile.

Profiles are either assigned by the system security administrator or may be a standard default profile that defines the lowest level of access to be granted a user. A privilege is the right to execute a particular function. Oracle has two types of privileges–system and object. They are loosely akin to the read, write, execute, delete, alter, or control access functions referred to earlier but are substantially more robust. Roles group several privileges together so that they can be granted or revoked simultaneously to a specific "class" of users.

Backup and Recovery

Off site storage of critical files is a requirement that must be documented and accomplished on a scheduled periodic basis. Storage of these critical records must be in a location different than the primary location of the database server. This may be accomplished through a communications link to the backup site or through the transfer of physical media. The files consist of sufficient data to restart the system at a different location or rebuild the system on a new computer at the current location.

The physical protection of the hardware, software, and other operational components will be under the control of ETSD and will be the same as the protection provided for any of the other servers maintained by ETSD.

An alternate site is to be identified at which the system can be reinitiated and communications established to the users.

Operational Responsibility

Basically there is little or no change to present operational responsibility of the AIRS-AQS subsystem. The mainframe is being exchanged for a client/server platform located in the EPA's National Computer Center (NCC). The anticipated operational responsibilities are:

- ETSD acquires the hardware and provides the facility space.
- ETSD provides system administration functions for the server.
- ETSD supports the communications requirements of the system.
- ETSD provides help desk support for the system infrastructure.
- ETSD provides system software maintenance.
- OAQPS is responsible for the maintenance of the application software, database design, and the application security.
- OAQPS provides help desk support for the application.
- OAQPS provides application training.

Server Hardware and System Software Requirements

For costing purposes, a Digital Equipment Corporation Alpha model 8400 5/300 server system has been evaluated as a potential platform for the AIRS-AQS database server. This is a dual processor system that can provide the expected performance and meet the capacity requirements of the AIRS-AQS subsystem. The system is easily expandable and upgradeable to meet future changing needs of AIRS-AQS. It is estimated that 10 GigaBytes of disk storage will be required with the system; 5 GB for working storage, 2.5 GB for system and application software, and 2.5 GB for expansion.

In addition to the Digital Unix operating system that is bundled with the Alpha server, it will be necessary to purchase the Oracle RDBMS kernel software, parallel query option, 64 bit option, and SQL*Net TCP/IP. All of these products must be licensed for 160 concurrent users.

An equivalent server configuration will be required for development of the AIRS-AQS subsystem and subsequent maintenance and support activities. The system does not have to provide the same performance and capacity as the production system but it should have identical system software configurations. Sofware licenses for 16 concurrent users should be sufficient for all of the system software.

Client Hardware and System Software Requirements

In order to fully utilize the potential of the new application the user will be required to interface with the application using a Personnel Computer (PC). This new workstation will have to meet the configuration requirements listed below. In most cases it is anticipated that equipment of this level exists in the inventory available to most of the users and that the expenses incurred would be limited to software requirements and minor workstation upgrades.

The minimum workstation hardware configuration is:

- 486 / DX33 processor
- 8 MB of Random Access Memory (16MB strongly recommended)
- 150 Megabytes of hard drive storage space
- VGA video card
- Mouse
- TCP/IP network connectivity

The minimum workstation software configuration is:

- Microsoft Windows v3.1 or Windows for Workgroups v3.11
- Oracle SQL*NET
- Oracle Discoverer/2000
- AIRS-AQS application software

Network Capacity Requirements

Network requirements should remain nearly the same as it is today. There may be some increase in traffic due to new capabilities that will be added. The design of the system will be focused on minimizing the impact on network traffic. Present reports and forms (screens) will be formatted at the workstation and will reduce this portion of the data flow over the network

It is not clear what the level of traffic of the current system is or how much capacity it uses. Communication costs are based upon a percentage of mainframe utilization costs. This figure is then used to allocate the communications costs to the application system. At present, the communication costs are bundled together and then distributed to three major areas - the mainframe, desktop systems, and distributed systems. What the allocations will be for a new "distributed" system is not known but may in fact be less than present allocations. ***It should be noted that allocations for communication costs are not based upon traffic.***

System Data Architecture

The purpose of this section is to document the high-level database design and dataflows that were determined during the development of this System Management Plan. The database design and dataflow will be further refined during the detailed system design phase of the project. They are used at this stage as an aid in understanding of the complexity of the legacy system and to provide some metrics that can be used for estimating system development costs and hardware costs based upon estimated capacity and performance requirements.

Database Design

The AIRS-AQS database is the focal point for all AQS user activity. It is imperative that the design of the database represents an accurate model of the information needs of the AIRS-AQS users. The database is used during the verification and validation processes for loading new data. It then provides data search and retrieval functions for report generation.

The primary entities that have been identified during the initial requirements analysis are described below. *Figure 4*, is an Entity Relationship Diagram that identifies the entities of the AIRS-AQS database and how they are related. The diagram provides a high-level view of the logical structure of the database as it currently exists in the mainframe environment.

Site - describes the location where one or more monitors are operated to obtain air quality sample data. It contains site information including state, county, site identifier, latitude/longitude, date established, land use, location description, and supporting Agency.

Monitor - contains detailed information about air quality monitoring stations. Information such as the parameter being measured, monitor type, effective date, reporting organization, measurement scale, and the date of the last EPA audit is maintained.

Hourly Sample - contains values for samples that have a sampling interval of less than twenty-four hours, with the most common interval being one hour. Typically, these values are measured by automated equipment that continuously measures the pollutant concentration and derives an average value for the sampling period.

Daily Sample - contains values for samples that have a sampling interval of twenty-four hours or more, with the most common interval being twenty-four hours. The values are usually derived from the chemical analysis of a discrete object such as a filter paper or a reagent solution that has been exposed to the ambient air for the duration of the sampling period.

Composite Sample - contains values derived from multiple samples that were combined and analyzed as a single data set. The time period represented by the composite can vary between an annual composite to a weekly composite. Monthly and quarterly composites are the most frequent types. It contains information that identifies the monitor the samples were recorded by, the time period represented by the value, information about how many samples were included in the composite sample, the method used, and the actual data values used to calculate the composite value.

Sample Summary - contains calculated values of concentrations of a parameter (pollutant) sampled at a site and summarized on a quarterly and yearly basis. Information such as the number of observations, monitor type, method of collection and analysis, and arthimetic mean is recorded for each summary entry.

SLAMS Annual Pollutant Summary - contains the state and local air monitoring report data for the various pollutants being measured. Information about the source and type of data, number of observations, and reported pollutant concentration values are recorded.

Monitor History - contains information that describes changes to site and monitor data. Most site-level changes are considered corrections and do not generate a history transaction. History transactions are generated for changes to most of the monitor data.

Precision and Accuracy Monitor Data - contains values that identify the precision and accuracy of the measurement process that is used to obtain the measurement data. Precision assessments are determined by performing repetitive measurements of ambient-level calibration gases at two-week intervals for continuous analyzers or by obtaining duplicate results from co-located monitors for manual methods. The accuracy assessments indicate the agreement between an analyzer measurement and a known audit standard concentration for continuous analyzers or the agreement between an observed value and a reference value for manual methods.

Precision and Accuracy Reporting Organization Summary - contains information about the collective or average precision and accuracy of groups of similar monitors operated by a specific reporting organization. The quarterly and yearly summary information is recorded for each monitor.

Precision and Accuracy Monitor Summary - contains information about the collective or average precision and accuracy of selected monitors. The quarterly and yearly summary information is recorded for each monitor.

Screening Data - contains information that is in the process of being verified and validated. Validation reports run against this data to identify errors or potential errors in the data. The corrections are made to this data and it is then transferred into other entities of the database as validated and verified information.

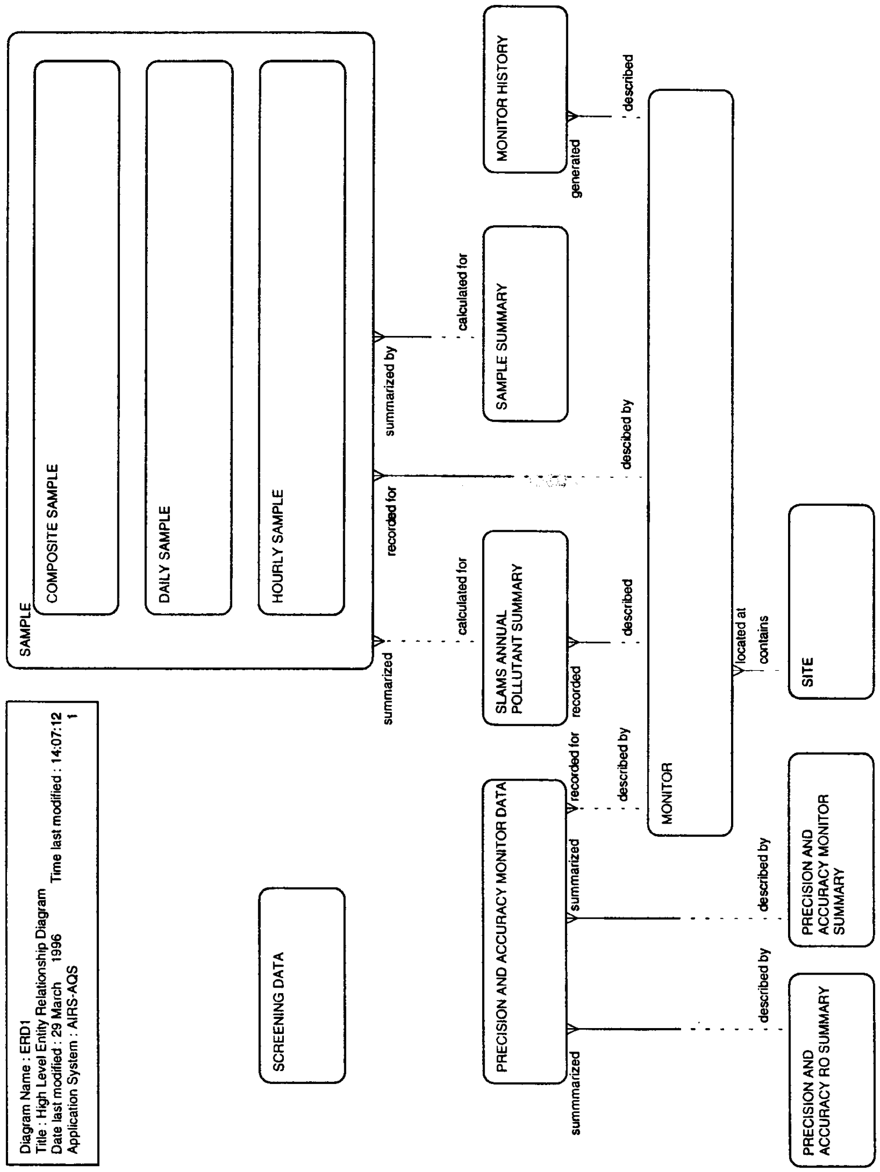


Figure 4. ***AQS Entity Relationship Diagram***

5.0 Cost/Benefits Analysis

In order to perform a cost/benefit analysis it is necessary to estimate the operating costs for the existing mainframe AIRS-AQS subsystem and estimate what the costs for the re-engineered AIRS-AQS will be. The estimates for the re-engineered AIRS-AQS based upon the high-level requirements that have been identified. The cost estimates will be further refined and validated as the detailed requirements are identifed in subsequent phases of the project.

Estimated Expenditures for Mainframe AIRS-AQS

The procedures used by ETSD to allocate WCF mainframe computing costs to clients are based upon CPU utilization by National System Code (NSC). This method does not provide sufficient granularity to estimate the cost of AIRS-AQS, since the applicable NSC of "AIRS" includes all subsystems (AFS, AQS, GCS, and AGS). Therefore, an alternative method had to be developed to estimate the charges directly related to AIRS-AQS usage.

Several assumptions were made to calculate the costs within a reasonable range of accuracy. The assumptions were:

- The OAQPS and ETSD staff have made a good faith effort in providing reasonable operating cost information for the existing mainframe AIRS-AQS subsystem.
- Total operating costs for AIRS are available and include the operating costs for the AIRS-AQS subsystem.
- A relationship can be established between the total AIRS operating costs and the proportion of those costs directly related to AIRS-AQS system usage.
- Most user needs are satisfied within a single mainframe application and users do not typically switch between mainframe applications.
- For the purposes of this cost/benefit analysis the costs do not have to be exact but should be a best estimate.
- Errors that may be generated by the cost calculations(estimates) are limited to the assignment of "AIRS mainframe account" data among AIRS subsystems. The accounts capturing the largest usage of mainframe CPU time are likely to be correctly assigned to subsystems due to the institutional knowledge of OAQPS and ETSD staff. Therefore, uncertain assignments are confined to a small fraction of AIRS CPU utilization.

The estimated annual AIRS-AQS expenditures for seven (7) fiscal years are shown in *Table 4. Estimated Expenditures for Mainframe AIRS-AQS.* The values shown in the table are based solely upon use of the mainframe-based AIRS-AQS subsystem. The process used for determining the estimated AIRS-AQS expenditures required mainframe usage data to be analyzed for the total

AIRS account activity[12]. The total AIRS-AQS operating costs were estimated to be 35% of the total AIRS operating costs based upon the AIRS-AQS account activity.

The WCF Expenditures section of the table shows the estimated expenditures attributable to AIRS-AQS system usage. These expenditures have traditionally been included in the monthly WCF Billing Statements submitted to OAQPS by ETSD. The expenditures have been broken down into processing expenditures and communication expenditures.

The estimated AIRS-AQS Processing Expenditures for FY96 are based upon a WCF Billing Statement for the period from October 1, 1995 through January 31, 1996. These charges cover the costs of providing mainframe computing services and includes facilities, supplies, maintenance, system support services, and other miscellaneous costs. The expenditures are 60%[13] of the WCF Billing Statement charges that are attributable to AIRS-AQS. The estimated AIRS-AQS processing expenditures for FY97 to FY02 have been allocated a 10% annual increase[14] to cover growth and general cost increases.

The AIRS-AQS Communication Expenditures for FY96 are also based upon a WCF Billing Statement for the period from October 1, 1995 through January 31, 1996. These charges cover the cost of providing communication services that are needed to support access to the NCC computing resources. The expenditures are the remaining 40% of the WCF Billing Statement charges that are attributable to AIRS-AQS. The estimated AIRS-AQS communications expenditures for FY97 to FY02 have been allocated a 10% annual increase to cover growth and general cost increases.

The OAQPS Budget Expenditures section of the table shows the estimated expenditures that OAQPS will encounter annually to support the mainframe-based AIRS-AQS subsystem. The Annual Maintenance Expenditures are paid by OAQPS from its annual operating budget. These charges cover the costs associated with maintaining a staff that provides system maintenance, help desk support and software problem resolution. The estimated AIRS-AQS annual maintenance expenditures for FY97 to FY02 have been allocated a 10% annual increase to cover growth and general cost increases.

The Total Annual Expenditures shows the estimated total cost of providing AIRS-AQS services and is the sum of the Total WCF Expenditures and Total OAQPS Budget Expenditures.

---

12 Information in these data files included NSCs, account numbers, FIMAS codes, user ID's, and CPU utilization information. A cross-reference file was used to correlate the NSCs with account numbers used by AIRS clients. Total AIRS operating costs were calculated from the association of account numbers with the AIRS NSC. These records were then further analyzed by FIMAS code to remove records that clearly identified non-AQS subsystems of AIRS. The remaining records were assumed to be related to AIRS-AQS processing and were used to calculate the total operating costs allocated to AIRS-AQS.

13 ETSD estimates that 40% of the mainframe processing charges are attributed to communication services.

14 The forecasted growth rate is documented in Applications Resources Usage Report (deliverable ITAS8-330 4-1) dated October 13, 1995.

Table 4. Estimated Expenditures for Mainframe AIRS-AQS

	FY96	FY97	FY98	FY99	FY00	FY01	FY02
WCF Expendituress							
AIRS-AQS Processing Expenditures	537,481	591,229	650,352	715,387	786,926	865,618	952,180
AIRS-AQS Communication Expenditures	358,320	394,152	433,567	476,924	524,616	577,078	634,786
Total WCF Expenditures	895,801	985,381	1,083,919	1,192,311	1,311,542	1,442,696	1,586,966
OAQPS Budget Expenditures							
Annual Maintenance Expenditures	240,000	264,000	290,400	319,440	351,384	386,522	425,174
Total OAQPS Budget Expenditures	240,000	264,000	290,400	319,440	351,384	386,522	425,174
Total Annual Expenditures	**1,135,801**	**1,249,381**	**1,374,319**	**1,511,751**	**1,662,926**	**1,829,218**	**2,012,140**

Estimated Expenditures for Re-engineered AIRS-AQS

The AIRS-AQS re-engineering project requires OAQPS to make an investment in developing the new system while continuing to incur budget expenditures for the existing mainframe AIRS-AQS subsystem for approximately 24 months. When the re-engineered AIRS-AQS is implemented in April, 1998 all users will start using the new system and all mainframe processing expenditures will stop being incurred.

Several assumptions were made to enable the re-engineered AIRS-AQS costs to be calculated and provide a reasonable accuracy range. The assumptions made are:

- A good faith effort has been made in providing reasonable expenditures estimates for the re-engineered AIRS-AQS subsystem.
- The expenditures for FY98 have been pro-rated based upon an anticipated April 1998 startup.
- The estimated expenditures have not taken into account any costs for implementing the AIRS Graphics Subsystem (AGS) on a client/server platform.
- The estimated WCF processing expenditures have been determined using ETSD's rule of thumb that client/server costs are 25% to 33% less than mainframe processing costs.
- The estimated expenditures do not include any required upgrades to client workstations since no formal survey has been conducted to document any upgrade requirements.
- It is estimated that the software required for each client workstation will cost between

$500 to $1,000, however, these costs have not been included in the estimated expenditures.

- The hardware and software costs are based upon vendor list prices and do not reflect any discounts that may be applicable to the EPA.
- For the purposes of this cost/benefit analysis the costs do not have to be exact but should be a best estimate.

The estimated annual AIRS-AQS expenditures for the seven (7) fiscal years are shown in the *Table 5. Estimated Expenditures for Re-engineered AIRS-AQS.* The values shown in the table are based upon the development of the AIRS-AQS subsystem with startup of the new system in April, 1998.

The WCF Expenditures section of the table shows the estimated expenditures attributable to re-engineered AIRS-AQS system usage. These expenditures will be included in the monthly WCF Billing Statements submitted to OAQPS by ETSD. The expenditures have been broken down into processing expenditures and communication expenditures.

The estimated AIRS-AQS Processing Expenditures reflect the estimated costs that will be charged under the WCF for daily operation of the re-engineered AIRS-AQS subsystem. These costs include operational hardware/software procurement, equipment maintenance, system software maintenance, media supplies, facilities charges, and part-time support labor for system administration. The operational hardware expenditures were determined by obtaining vendor pricing for the equipment evaluated for implemention of the re-engineered AIRS-AQS subsystem. The total estimated hardware costs ($254,002) were amortized over a 36 month period that spans four fiscal years (FY98 to FY01).

The operational hardware will be procured in January, 1998, three months prior to implementation of the system. The estimated operational hardware expenditures for FY99 to FY02 have been allocated an additional 10% of the original hardware cost ($25,400) to cover growth and general cost increases. The Operational Software Expenditures were determined by obtaining vendor pricing for the system software evaluated for implemention of the re-engineered AIRS-AQS subsystem. The total estimated hardware costs ($375,410) were amortized over a 36 month period that spans four fiscal years (FY98 to FY01).

The operational software will be procured in January, 1998, three months prior to implementation of the system. The estimated operational software expenditures for FY99 to FY02 have been allocated an additional 10% of the original software cost ($37,541) to cover growth and general cost increases. The estimated processing expenditures shown in the table for FY98 are 30% less[15] than the FY98 processing expenditures for the mainframe AIRS-AQS. The AIRS-AQS processing expenditures for FY98 have been prorated at 60% of the estimated annual expenditure based upon system startup in April, 1998. The annual processing expenditures have been allocated a 10% annual increase to cover growth and general cost increases.

The estimated AIRS-AQS Communication Expenditures refelect the estimated communcations

15 ETSD has stated that, as a rule of thumb, Unix client/server costs are 25% to 33% less than the equivalent mainframe costs.

costs that will be charged under the WCF. These charges cover the cost of providing communication services that are needed to support access to the NCC computing resources. It is generally accepted that the charges for communication services in the client/server environment will be approximateley the same as in the mainframe environment. Therefore, the expenditure estimates shown in *Table 5. Estimated Expenditures for Re-engineered AIRS-AQS* are the same estimates that were include in *Table 4. Estimated Expenditures for Mainframe AIRS-AQS.* The AIRS-AQS communication expenditures for FY98 have been prorated at 40% of the estimated annual expenditure based upon system startup in April, 1998. The estimated AIRS-AQS communications expenditures for FY97 to FY02 have been allocated a 10% annual increase to cover growth and general cost increases.

The OAQPS Budget Expenditures section of the table shows the estimated expenditures that OAQPS will encounter annually to support the re-engineered AIRS-AQS subsystem.

The OAQPS Application Development Expenditures show the costs that OAQPS will incur for development of the re-engineered AIRS-AQS subsystem. The application development will span a twenty-four month period over three fiscal years (FY96 to FY98). The costs are based upon the deliverables that will be provided as descibed in the Life Cycle Methodology. The software development project team will be composed of six (6) personnel that will be applied in various loading capacities during the system development life cycle. The project team will be composed of one project manager, four systems analysts/programmers, and one database administrator/data analyst.

The Development Hardware Expenditures shows the investment that OAQPS will make to provide the computer equipment needed to re-engineer the AIRS-AQS subsystem and support the subsystem after implementation. This equipment provides a method for making changes to the application with minimal impact on the operation of the production system. The equipment to be purchased includes a Unix server and Pentium workstations for the development team. The estimated expenditures were determined by obtaining vendor pricing for the equipment evaluated for development of the re-engineered AIRS-AQS subsystem. The total estimated hardware costs ($109,898) were amortized over a 36 month period that spans four fiscal years (FY96 to FY99). The development hardware will be procured in June, 1996 so that it can be used as the CASE repository during the detailed analysis and design phases. The estimated development hardware expenditures for FY97 to FY02 have been allocated an additional 10% of the original hardware cost ($10,990) to cover growth and general cost increases and an additional 15% of the original hardware cost ($16,485) to cover annual equipment maintenance costs.

The Development Software Expenditures shows the investment that OAQPS will make to provide the system software needed to re-engineer the AIRS-AQS subsystem and support the system after implementation. The development software includes the Unix operating system software, Oracle RDBMS, Designer/2000, Developer/2000 and Discoverer/2000. This software is state-of-the-art which are designed to support the rapid development methodology described previously and promote consistency and uniformity during the development process. They also allow for close monitoring and ease the coordination of the developers individual efforts. The estimated expenditures were determined by obtaining vendor pricing for the software evaluated for development of the re-engineered AIRS-AQS subsystem. The total estimated software costs ($70,909) were amortized over a 36 month period that spans four fiscal years (FY96 to FY99). The development software will be procured in June, 1996 so that it can be used during the detailed analysis and design phases. The estimated development software expenditures for FY97 to FY02 have been allocated an additional 10% of the original software cost ($7,091) to cover growth and general cost increases and an additional 15% of the original software cost ($10,636) to cover annual software support and maintenance costs.

The Annual Maintenance Expenditures are paid by OAQPS from its annual operating budget. These charges cover the costs associated with maintaining a staff that provides system maintenance, help desk support, and software problem resolution. For the first year after implementation of the re-engineered AIRS-AQS, the entire development staff (6 personnel) will be retained in a maintenance/support capacity to train users, support the system, correct software problems, and make any needed enhancements to the system. After that time the support staff should be reduced to four (4) full-time personnel for one year and then be reduced to three (3) full-time personnel for continuing years. The estimated maintenance expenditures for FY98 have been prorated at 50% of the annual expenditure ($330,000) based upon startup of the re-engineered AIRS-AQS in April, 1998. The estimated AIRS-AQS annual maintenance expenditures for FY97 to FY02 have been allocated a 10% annual increase to cover growth and general cost increases.

The Total Annual Expenditures shows the estimated total cost of providing re-engineered AIRS-AQS services and is the sum of the Total WCF Expenditures and Total OAQPS Budget Expenditures.

Table 5. Estimated Expenditures for Re-engineered AIRS-AQS

	FY96	FY97	FY98	FY99	FY00	FY01	FY02
WCF Expenditures							
AIRS-AQS Processing Expenditures			430,495	773,408	823,585	692,492	729,467
AIRS-AQS Communications Expenditures			260,140	476,924	524,616	577,078	634,786
Total WCF Expenditures			690,635	1,250,432	1,348,201	1,298,425	1,364,253
OAQPS Direct Budget Expenditures							
Application Development Expenditures	101,442	400,290	208,500				
Development Hardware Expenditures	12,212	64,111	64,111	51,889	27,475	27,475	27,475
Development Software Expenditures	7,880	41,361	41,361	33,470	17,721	17,721	17,721
Annual Maintenance Expenditures			165,000	308,000	338,800	372,680	409,948
Total OAQPS Direct Budget Expenditures	121,534	505,762	478,972	393,359	383,996	417,876	455,144
Total Annual Operating Expenditures	121,534	505,762	1,169,607	1,643,791	1,732,197	1,716,301	1,819,397

Summary

This cost/benefit summary is based upon estimated expenditures that could be incurred by EPA to continue operating the mainframe AIRS-AQS subsystem or develop and implement the re-engineered AIRS-AQS. One of the considerations to keep in mind while reviewing this information is the fact that detailed data on actual costs is difficult to obtain, especially when looking at a subsystem (AQS) that is part of a larger system (AIRS).

The graph shown in *Figure 5. Comparison of WCF and OAQPS Budget Expenditures* compares the estimated expenditures for remaining on the mainframe (Mainframe AQS) to a client/server system (Re-engineered AIRS-AQS). The graph includes the expenditures for developing the re-engineered AIRS-AQS (during FY96 to FY98) and continued maintenance support for the system. The graph indicates that OAQPS budget expenditures are larger during the development period but then level off and are close to the estimated expenditures for the mainframe AIRS-AQS. The WCF expenditures for the Re-engineered AIRS-AQS are slightly higher than the mainframe AQS expenditures during FY99 and FY00 due to amortization of the operational hardware and software. However, beginning in FY01 the benefits of the re-engineered AIRS-AQS begin to be realized by having significantly lower WCF expenditures.

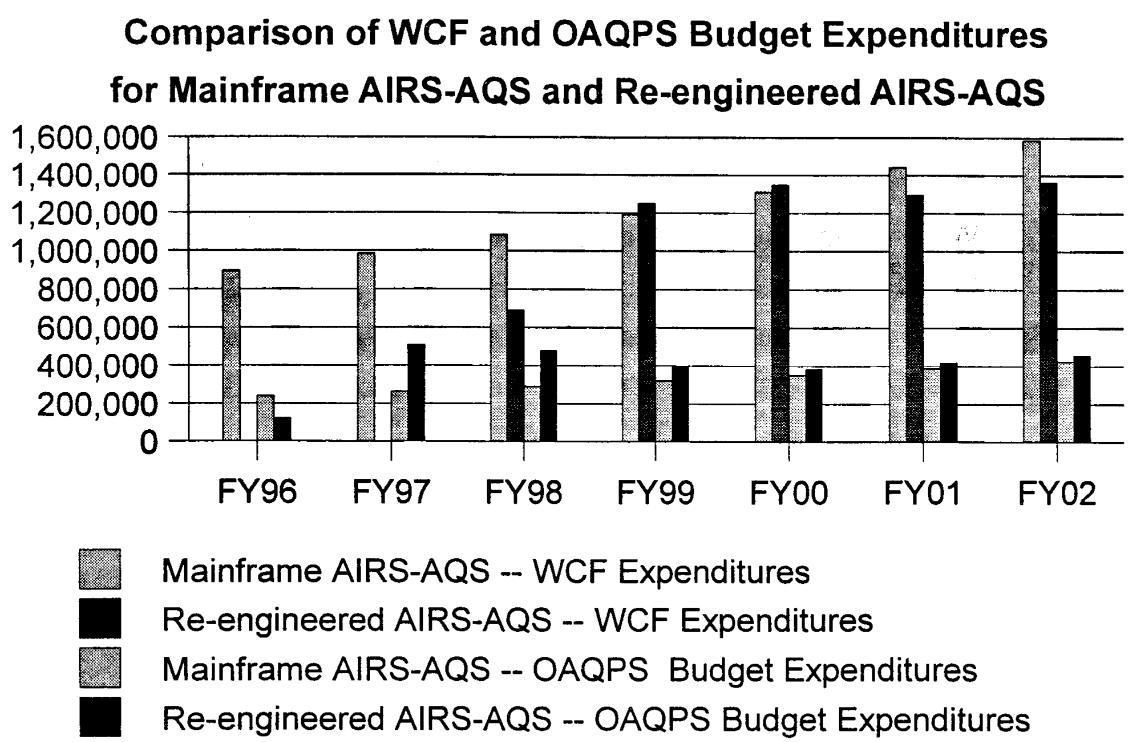


Figure 5. ***Comparison of WCF and OAQPS Budget Expenditures***

The graph shown in *Figure 6,* depicts the estimated annual expenditures for the Re-engineered AIRS-AQS subsystem. The graph includes the portion of the mainframe AIRS-AQS expenditures that will be incurred until implementation of the re-engineered system in April, 1998 to show the total budgetary impact associated with re-engineering AIRS-AQS.

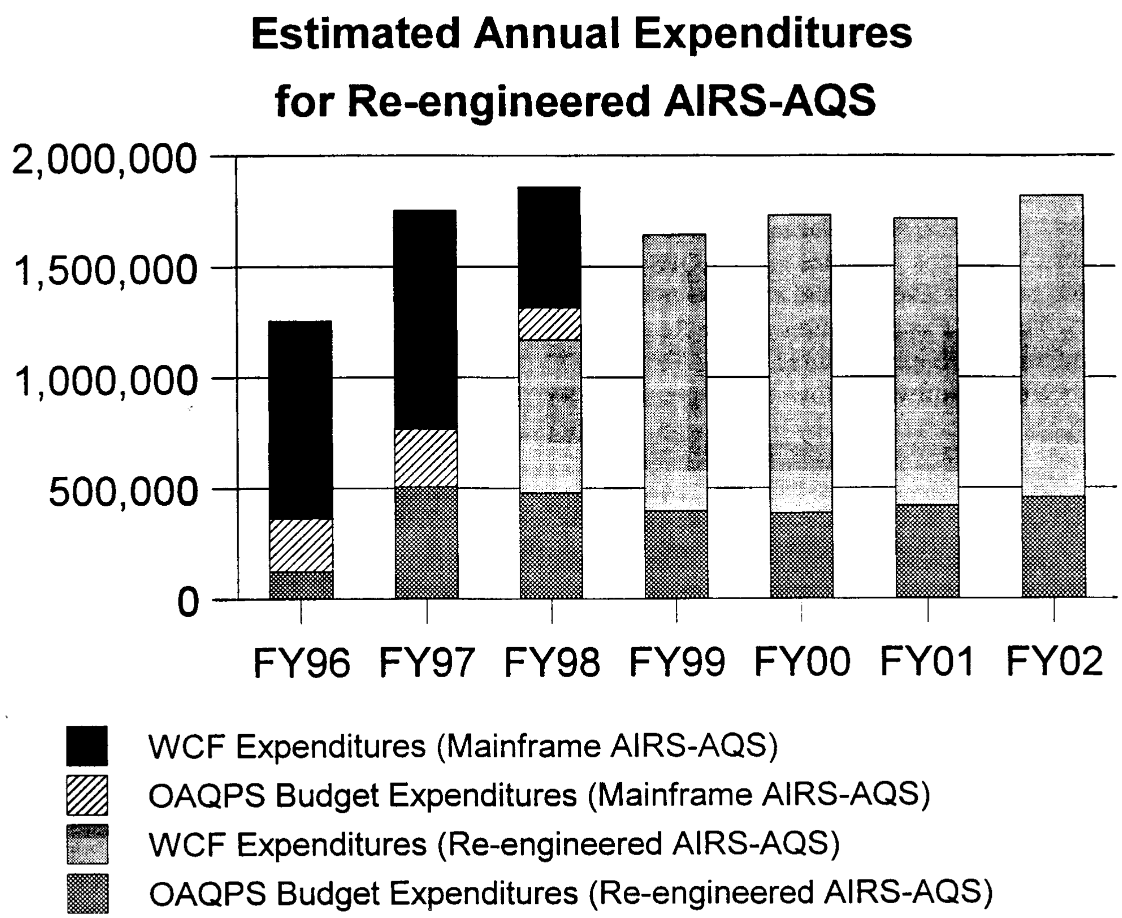


Figure 6. ***Estimated Annual Expenditures***

The graph shown in *Figure 7*, depicts the estimated annual expenditures for continued use of the mainframe AIRS-AQS subsystem.

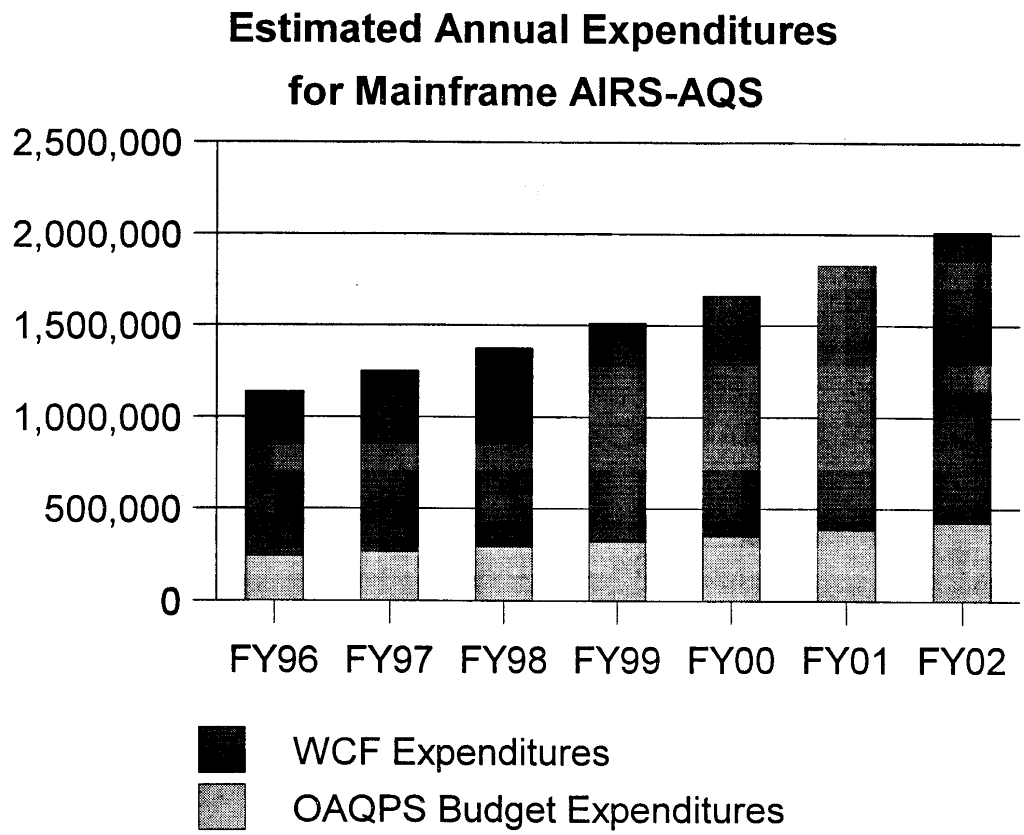


*Figure 7. **Estimated Annual Expenditures***

Based upon these estimated annual expenditures, the graph in *Figure 8*, shows the annual cost savings that can be expected by implementing the re-engineered AIRS-AQS subsystem and moving the susbsystem off the mainframe. As depicted in the graph, no monetary savings are realized until FY01 due to development costs and computer hardware/software purchase costs. However, beginning in FY01 the savings in WCF expenditures start to accrue. These savings in WCF expenditures will become important to OAQPS as the WCF becomes operational.

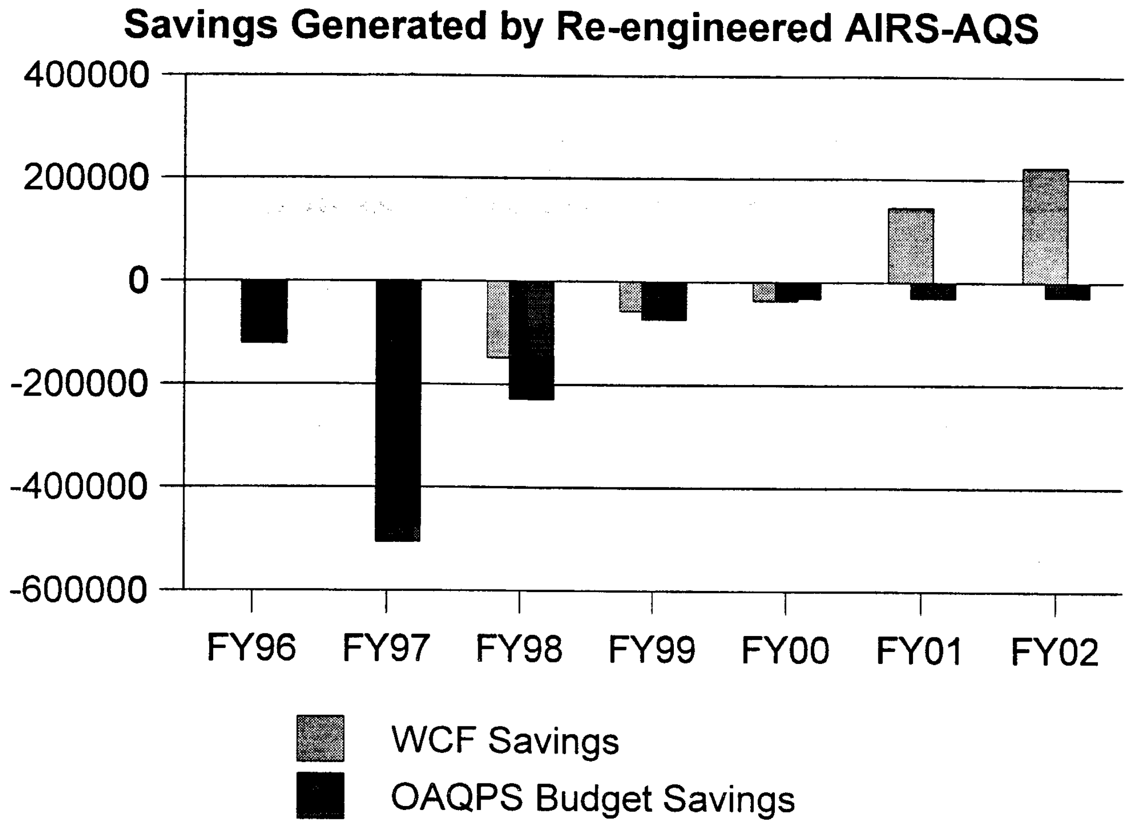
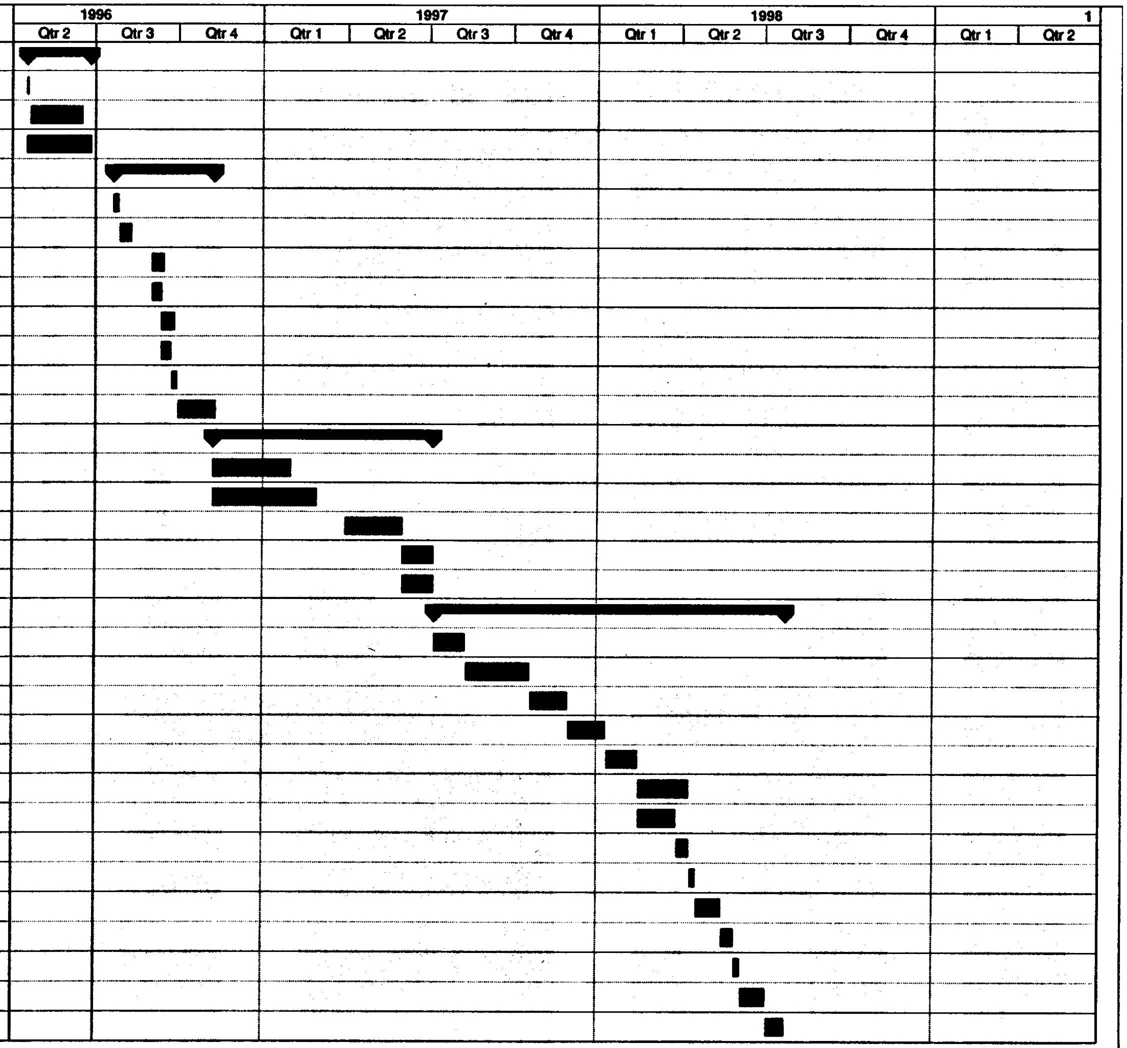


Figure 8. ***Savings Generated by Re-engineered AIRS-AQS***

It is also important to note that the savings depicted in the graph do not reflect the increased functionality that will be obtained with the re-engineered AIRS-AQS subsystem. Other intangible benefits (increased productivity, flexibility, and data integrity), which are difficult to place a monetary savings on, will also be created by the new system.

Appendix A. Detailed Project Management Plan

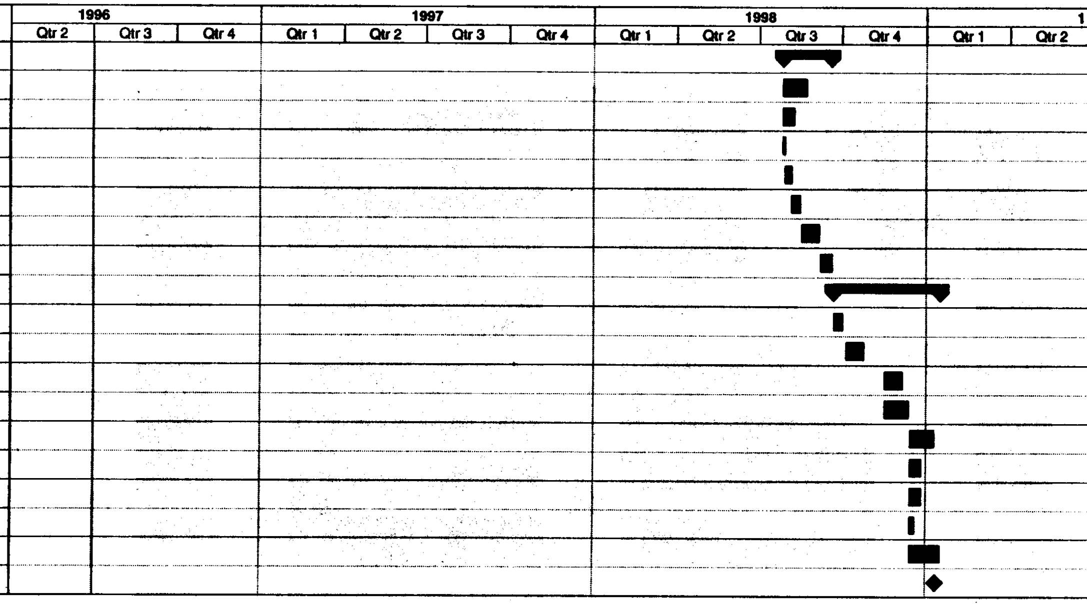
ID	Task Name
1	**Project Initiation**
2	identify team members
3	review project plan and expand tasks
4	prepare report and briefing
5	**Concept And Analysis**
6	order all development hardware and software
7	install development hardware and software
8	review questionnaire results
9	review existing systems documentation
10	prepare detailed system requirements
11	finalize security policies and procedures
12	meet with user advisory group
13	develop demonstration prototype
14	**Database Definition And Design**
15	prepare data flow diagram
16	prepare database design diagrams
17	systems interfaces design
18	mapping of new to existing system
19	design communications interfaces
20	**Database Development**
21	prepare information model
22	design & create database and interfaces
23	design & create user interface
24	design & create standard reports
25	creations of load programs
26	create batch update programs
27	create browser views
28	create home-page interface (Internet)
29	load database subset
30	construct & install the application object
31	perform integration test
32	review test results
33	system modifications and corrections
34	finalize system interfaces



Project:
Date: 4/1/96

Task | Progress | Milestone | Summary | Rolled Up Task | Rolled Up Milestone | Rolled Up Progress

ID	Task Name
35	**Acceptance Test**
36	implement security system
37	prepare draft user documentation for test
38	select test site
39	create application test files
40	perform test
41	test evaluation and system modifications
42	application acceptance
43	**Roll Out Preparation**
44	finalize roll out process
45	finalize system documentation
46	finalize user documentation
47	develop training program
48	conduct user training sessions
49	Install production database
50	install production software
51	setup user support interface(help desk)
52	prepare notices to user community
53	**Cut-over and Initial Support**



Project:
Date: 4/1/96

Task | Progress | Milestone | Summary | Rolled Up Task | Rolled Up Milestone | Rolled Up Progress